Programming Guide RIGOL

Publication Number: PGA04105-1110
Oct. 2008

DS1000B Series Digital Oscilloscope

DS1204B/DS1104B/DS1064B

© 2008 RIGOL Technologies, Inc. All Rights Reserved

- © 2008 RIGOL Technologies, Inc. All Rights Reserved
- RIGOL products are protected by patent law in and outside of P.R. China.
- Information in this publication replaces all previously corresponding material.
- RIGOL Technologies, Inc. reserves the right to modify or change part of or all the specifications and pricing policies at company's sole decision.

NOTE: RIGOL is registered trademark of **RIGOL** Technologies, Inc.

Content

Chapter 1 Programming Introduction 1-1

Communication Ports 1-2
Symbol Instruction 1-3
Command Abbreviation 1-4

Chapter 2 DS1000B Command Systems 2-1

System Commands 2-2
Keyboard Commands 2-6
Measurement Commands 2-22
Acquire Commands 2-33
Display Commands 2-36
Vertical Commands 2-42
Horizontal Commands 2-49
Trigger Commands 2-54
Trigger Control Commands 2-54
Trigger Mode Commands 2-60
Math Commands 2-77
Waveform Commands 2-79
Utility Commands 2-88
Storage Commands 2-91
Mask Commands 2-96
Cursor Commands 2-102
Other Commands 2-109

Chapter 3 Programming Examples 3-1

Example 1 The Preparation for Programming 3-2
Example 2 Setting the basic measurement 3-3
Example 3 Storaging and Setting the File 3-4
Example 4 Reading the Data of Screen Image 3-6

Command Quick Reference I

Chapter 1 Programming Introduction

This book provides guidance to the remote control programming of the DS1000B series digital oscilloscopes.

It includes the necessary conceptual information, the command structure for the subsystems and the application cases. Users can carry out the remote control to the oscilloscope or do the further development according to this manual.

NOTE: The Programming Guide is applicable to the firmware version of the Oscilloscope above 00.02.04.

Communication Ports

Computers can communicate with the oscilloscope by sending and receiving messages over USB and LAN interface. Commands in the form of ASCII character strings are embedded in your computer to make control easier. The input commands of the host language are used to read responses from the oscilloscope.

Basic operations that you can do with a computer and an oscilloscope include:

- Set up the oscilloscope;
- Relational measurements;
- Acquire data (waveforms or measurement data) from the oscilloscope.

Symbol Instruction

1. Command Sperator :

Commands usually begin with a colon (:), in special case, such as IEEE commands will begin with (*), it represents operations the oscilloscope will take. Queries are formed by adding a question mark (?) at the end of the commands. The return values of the queries are mostly expressed in scientific notation. Forms of commands and inquiries have different programme datum usually. But some others are not need to input the programme datum.

2. Blank Space □

Blank space is used to separate the instruction from the program data. If the command does not include any program data, a blank space is not needed.

3. Braces { }

If several options separated by "|" are enclosed via "{ }", only one element may be selected. For example, {ON | OFF} indicates that only ON or OFF may be available, not both.

4. Square Brackets [ ]

Items enclosed in "[ ]" can be chosen or not, it's optional.

5. Triangle Brackets < >

Items enclosed in "< >" should be replaced by a numerical value or character string.

Command Abbreviation

Owing to some of the commands are too long to memory and spell, the commands system of DS1000B especially setup the Commands Abbreviation in order to reduce your workload and the difficulty to memory.

The abbreviation rule is all the small letters in the command header are omitted.

For example: :TRIGger:ALTernation:SOURce
Its abbreviation is:
:TRIG:ALT:SOUR

Chapter 2 DS1000B Command Systems

DS1000B Commands System includes the following command subsystems:

- System Commands
- Keyboard Commands
- Measurement Commands
- Acquire Commands
- Display Commands
- Vertical Commands
- Horizontal Commands
- Trigger Commands
- Math Commands
- Waveform Commands
- Utility Commands
- Storage Commands
- Mask Commands
- Cursor Commands
- Other Commands

System Commands

System Commands are used for the basic operations of an oscilloscope. They can not only be used as an individual command but also in conjunction with other commands to form instructions.

System Commands include:

- *IDN?
- :RUN
- :STOP
- :AUTO
- :FORCetrig
- :Trig%50
- *RST
- *LRN?
- *OPC?
- :SINGLE

1. *IDN?

Command Format:

*IDN?

Function:

The command queries the company name, the oscilloscope model, the product serial and the software version.

Return Format:

Company name,, <model>, <serial>, <version>.

Example:

*IDN? Return Rigol Technologies, DS1204B, DS10000000, 00.02.04.

2. :RUN

Command Format:

:RUN

Function:

Execute this command, the oscilloscope will start waveform sampling working ,if to stop working, execute **:STOP** command again.

3. :STOP

Command Format:

:STOP

Function:

Execute this command, the oscilloscope will stop waveform sampling working , if to restart working, execute **:RUN** command again.

4. :AUTO

Command Format:

:AUTO

Function:

The command makes the oscilloscope tests all input waveforms and set the waveforms automatically to get the optimum conditions to display.

5. :FORCetrig

Command Format:

:FORCetrig

Function:

The command wil produce a trigger signal to force the oscilloscope trigger and to display a waveform when there is no suitable trigger condition. It is mainly applicable to the "Normal" and "Single" trigger modes.

6. :Trig%50

Command Format:

:Trig%50

Function:

The command sets the trigger level at the vertical midpoint of the signal amplitude.

7. *RST

Command Format:

*RST

Function:

The command resets the system.

8. *LRN?

Command Format:

*LRN?

Function:

The command queries the system settings. The query returns the data of system settings in the form of a self-defined character string which could be downloaded to do the same settings in the future.

9. *OPC?

Command Format:

*OPC?

Function:

The command queries whether the command operation has been completed.

Return Format:

The query returns 0 or 1. 1 means operation has been completed, 0 means not.

10. :SINGLE

Command Format:

:SINGLE

Function:

The command sets the trigger mode as Single trigger, means that collect a waveform when detect a trigger signal, then stop running.

Keyboard Commands

Keyboard Commands are used to control the keys and knobs on the operation panel of DS1000B.

Keyboard Commands include:

- :KEY:LOCK
- :KEY:RUN
- :KEY:AUTO
- :KEY:SINGLe
- :KEY:CH1
- :KEY:CH2
- :KEY:CH3
- :KEY:CH4
- :KEY:MATH
- :KEY:REF
- :KEY:F1
- :KEY:F2
- :KEY:F3
- :KEY:F4
- :KEY:F5
- :KEY:MNUoff
- :KEY:MEASure
- :KEY:CURSor
- :KEY:ACQuire
- :KEY:DISPlay
- :KEY:STORage
- :KEY:UTILity
- :KEY:MNUTIME
- :KEY:TrigMENU
- :KEY:Trig%50
- :KEY:TrigMODE
- :KEY:TrigFORCe
- :KEY:HELP
- :KEY:QUICKMEASure
- :KEY:QUICKPRINt
- :KEY:FUNC_Z
- :KEY:FUNC_INC
- :KEY:FUNC_DEC
- :KEY:CH1_VOLT_INC
- :KEY:CH1_VOLT_DEC
- :KEY:CH1_VOLT_Z
- :KEY:CH1_POS_INC
- :KEY:CH1_POS_DEC
- :KEY:CH1_POS_Z
- :KEY:CH2_VOLT_INC
- :KEY:CH2_VOLT_DEC
- :KEY:CH2_VOLT_Z
- :KEY:CH2_POS_INC
- :KEY:CH2_POS_DEC
- :KEY:CH2_POS_Z
- :KEY:CH3_VOLT_INC
- :KEY:CH3_VOLT_DEC
- :KEY:CH3_VOLT_Z
- :KEY:CH3_POS_INC
- :KEY:CH3_POS_DEC
- :KEY:CH3_POS_Z
- :KEY:CH4_VOLT_INC
- :KEY:CH4_VOLT_DEC
- :KEY:CH4_VLOT_Z
- :KEY:CH4_POS_INC
- :KEY:CH4_POS_DEC
- :KEY:CH4_POS_Z
- :KEY:TIME_INC
- :KEY:TIME_DEC
- :KEY:TIME_Z

- :KEY:TIME_POS_INC
- :KEY:TIME_POS_DEC
- :KEY:TIME_POS_Z
- :KEY:TRIG_LEVEL_INC
- :KEY:TRIG_LEVEL_DEC
- :KEY:TRIG_LEVEL_Z

1. :KEY:LOCK

Command Format:

:KEY:LOCK□{ ENABle | DISable }

Function:

The command enables and disables the function of Remote control on the keys of front panel.

Query Format:

:KEY:LOCK?

Return Format:

The query returns ENABLE or DISABLE.

Example:

:KEY:LOCK□ENABle Enable remote control on the keys of front panel .

:KEY:LOCK? Return ENABLE.

2. :KEY:RUN

Command Format:

:KEY:RUN

Function:

This command controls the running state of the oscilloscope. The oscilloscope will cutover between RUN and STOP when sending this command continually.

3. :KEY:AUTO

Command Format:

:KEY:AUTO

Function:

The command sets oscilloscope automatically to display the waveform to be optimal condition.

4. :KEY:SINGLe

Command Format:

:KEY:SINGLe

Function:

The command sets the trigger mode as Single trigger.

5. :KEY:CH1

Command Format:

:KEY:CH1

Function:

The command sets channel 1 and its menu on or off.

6. :KEY:CH2

Command Format:

:KEY:CH2

Function:

The command sets channel 2 and its menu on or off.

7. :KEY:CH3

Command Format:

:KEY:CH3

Function:

The command sets channel 3 and its menu on or off.

8. :KEY:CH4

Command Format:

:KEY:CH4

Function:

The command sets channel 4 and its menu on or off.

9. :KEY:MATH

Command Format:

:KEY:MATH

Function:

The command sets Math function amd its menu on or off.

10. :KEY:REF

Command Format:

:KEY:REF

Function:

The command sets reference waveform function and its menu on or off.

11. :KEY:F1

Command Format:

:KEY:F1

Function:

The command selects the first option in current menu. If the subordinate menu existing and its options are available, then these options could be selected circularly when send repeatly the command.

12. :KEY:F2

Command Format:

:KEY:F2

Function:

The command selects the second option in current menu. If the subordinate menu existing and its options are available, then these options could be selected circularly when send repeatly the command.

13. :KEY:F3

Command Format:

:KEY:F3

Function:

The command selects the third option in current menu. If the subordinate menu existing and its options are available, then these options could be selected circularly when send repeatly the command

14. :KEY:F4

Command Format:

:KEY:F4

Function:

The command selects the fourth option in current menu. If the subordinate menu existing and its options are available, then these options could be selected circularly when send repeatly the command

15. :KEY:F5

Command Format:

:KEY:F5

Function:

The command selects the fifth option in current menu. If the subordinate menu existing and its options are available, then these options could be selected circularly when send repeatly the command

16. :KEY:MNUoff

Command Format:

:KEY:MNUoff

Function:

The command sets menu display function on or off.

17. :KEY:MEASure

Command Format:

:KEY:MEASure

Function:

The command sets measurement function and its menu on or off.

18. :KEY:CURSor

Command Format:

:KEY:CURSor

Function:

The command sets cursor measurement function and its menu on or off. The cursor mode can be set by sending the command continually.

19. :KEY:ACQuire

Command Format:

:KEY:ACQuire

Function:

The command sets acquire menu on or off.

20. :KEY:DISPlay

Command Format:

:KEY:DISPLAY

Function:

The command sets display menu on or off.

21. :KEY:STORage

Command Format:

:KEY:STORage

Function:

The command sets storage menu on or off.

22. :KEY:UTILity

Command Format:

:KEY:UTILITY

Function:

The command sets utility menu on or off.

23. :KEY:MNUTIME

Command Format:

:KEY:MNUTIME

Function:

The command sets horizontal system and its menu on or off.

24. :KEY:TrigMENU

Command Format:

:KEY:TrigMENU

Function:

The command sets trigger menu on or off.

25. :KEY:Trig%50

Command Format:

:KEY:Trig%50

Function:

This command sets the trigger level at the vertical midpoint of the amplitude of trigger signal.

26. :KEY:TrigMODE

Command Format:

:KEY:TrigMODE

Function:

The command shifts the trigger mode among AUTO, NORMAL and SINGLE.

27. :KEY:TrigFORCe

Command Format:

:KEY:TrigFORCe

Function:

The command is used for forcing trigger.

28. :KEY:HELP

Command Format:

:KEY:HELP

Function:

The command sets the bulit-in help system on or off.

29. :KEY:QUICKMEASure

Command Format:

:KEY:QUICKMEASure

Function:

The command sets quick-measurement function on or off. And it could be set in Measurement menu.

30. :KEY:QUICKPRINt

Command Format:

:KEY:QUICKPRINt

Function:

The command prints and saves the screen.

31. :KEY:FUNC_Z

Command Format:

:KEY:FUNC_Z

Function:

The command selects the multifunction knob.

32. :KEY:FUNC_INC

Command Format:

:KEY:FUNC_INC

Function:

The command increases the offset of multifunction knob.

33. :KEY:FUNC_DEC

Command Format:

:KEY:FUNC_DEC

Function:

The command decreases the offset of multifunction knob.

34. :KEY:CH1_VOLT_INC

Command Format:

:KEY:CH1_VOLT_INC

Function:

The command decreases the vertical scale of channel 1.

35. :KEY:CH1_VOLT_DEC

Command Format:

:KEY: CH1_VOLT_DEC

Function:

The command increases the vertical scale of channel 1

36. :KEY:CH1_VOLT_Z

Command Format:

:KEY: CH1_VOLT_Z

Function:

The command sets the adjustment mode of vertical scale of channel 1 as Fine or Coarse. The vertical scale steps by 1-2-5 in Coarse; and by equality in Fine.

37. :KEY:CH1_POS_INC

Command Format:

:KEY: CH1_POS_INC

Function:

The command increases the vertical offset of channel 1 evenly.

38. :KEY:CH1_POS_DEC

Command Format:

:KEY:CH1_POS_DEC

Function:

The command decreases the vertical offset of channel 1 evenly.

39. :KEY:CH1_POS_Z

Command Format:

:KEY: CH1_POS_Z

Function:

The command adjusts the vertical offset of channel 1 to zero.

40. :KEY:CH2_VOLT_INC

Command Format:

:KEY:CH2_VOLT_INC

Function:

The command decreases the vertical scale of channel 2.

41. :KEY:CH2_VOLT_DEC

Command Format:

:KEY:CH2_VOLT_DEC

Function:

The command increases the vertical scale of channel 2.

42. :KEY:CH2_VOLT_Z

Command Format:

:KEY:CH2_VOLT_Z

Function:

The command sets the adjustment mode of vertical scale of channel 2 as Fine or Coarse. The vertical scale steps by 1-2-5 in Coarse; and by equality in Fine.

43. :KEY:CH2_POS_INC

Command Format:

:KEY:CH2_POS_INC

Function:

The command increases the vertical position of channel 2 evenly.

44. :KEY:CH2_POS_DEC

Command Format:

:KEY:CH2_POS_DEC

Function:

The command decreases the vertical position of channel 2 evenly.

45. :KEY:CH2_POS_Z

Command Format:

:KEY:CH2_POS_Z

Function:

The command adjusts the vertical offset of channel 2 to zero.

46. :KEY:CH3_VOLT_INC

Command Format:

:KEY:CH3_VOLT_INC

Function:

The command decreases the vertical scale of channel 3.

47. :KEY:CH3_VOLT_DEC

Command Format:

:KEY:CH3_VOLT_DEC

Function:

The command increases the vertical scale of channel 3.

48. :KEY:CH3_VOLT_Z

Command Format:

:KEY:CH3_VOLT_Z

Function:

The command sets the adjustment mode of vertical scaling of channel 3 as Fine or Coarse. The vertical scale steps by 1-2-5 in Coarse; and by equality in Fine.

49. :KEY:CH3_POS_INC

Command Format:

:KEY:CH3_POS_INC

Function:

The command increases the vertical offset of channel 3 evenly.

50. :KEY:CH3_POS_DEC

Command Format:

:KEY:CH3_POS_DEC

Function:

The command decreases the vertical offset of channel 3 evenly.

51. :KEY:CH3_POS_Z

Command Format:

:KEY:CH3_POS_Z

Function:

The command adjusts the vertical offset of channel 3 to zero.

52. :KEY:CH4_VOLT_INC

Command Format:

:KEY:CH4_VOLT_INC

Function:

The command decreases the vertical scale of channel 4.

53. :KEY:CH4_VOLT_DEC

Command Format:

:KEY:CH4_VOLT_DEC

Function:

The command increases the vertical scale of channel 4.

54. :KEY:CH4_VLOT_Z

Command Format:

:KEY:CH4_VOLT_Z

Function:

The command sets the adjustment mode of vertical scale of channel 4 as Fine or Coarse. The vertical scale steps by 1-2-5 in Coarse; and by equality in Fine.

55. :KEY:CH4_POS_INC

Command Format:

:KEY:CH4_POS_INC

Function:

The command increases the vertical offset of channel 4 evenly.

56. :KEY:CH4_POS_DEC

Command Format:

:KEY:CH4_POS_DEC

Function:

The command decreases the vertical offset of channel 4 evenly.

57. :KEY:CH4_POS_Z

Command Format:

:KEY:CH4_POS_Z

Function:

The command adjusts the vertical offset of channel 4 to zero.

58. :KEY:TIME_INC

Command Format:

:KEY:TIME_INC

Function:

The command decreases the time base by 1-2-5 step.

59. :KEY:TIME_DEC

Command Format:

:KEY:TIME_DEC

Function:

The command increases time base by 1-2-5 step.

60. :KEY:TIME_Z

Command Format:

:KEY:TIME_Z

Function:

The command sets delayed scan function on or off.

61. :KEY:TIME_POS_INC

Command Format:

:KEY:TIME_POS_INC

Function:

The command decreases the trigger offset to the horizontal zero point evenly.

62. :KEY:TIME_POS_DEC

Command Format:

:KEY:TIME_POS_DEC

Function:

The command increases the trigger offset to the horizontal zero point evenly.

63. :KEY:TIME_POS_Z

Command Format:

:KEY:TIME_POS_Z

Function:

The command adjusts the trigger offset to the horizontal zero point evenly.

64. :KEY:TRIG_LEVEL_INC

Command Format:

:KEY:TRIG_LEVEL_INC

Function:

The command increases the trigger level evenly.

65. :KEY:TRIG_LEVEL_DEC

Command Format:

:KEY:TRIG_LEVEL_DEC

Function:

The command decreases the trigger level evenly.

66. :KEY:TRIG_LEVEL_Z

Command Format:

:KEY:TRIG_LEVEL_Z

Function:

The command adjusts the trigger level to zero.

Measurement Commands

Measurement Commands are used for the fundamental measurement operations, and the measurement results are expressed by scientific notation.

Measurement Commands include:

- :MEASure:CLEar
- :MEASure:VPP?
- :MEASure:VMAX?
- :MEASure:VMIN?
- :MEASure:VAMPlitude?
- :MEASure:VTOP?
- :MEASure:VBASe?
- :MEASure:VAVerage?
- :MEASure:VRMS?
- :MEASure:OVERshoot?
- :MEASure:PREShoot?
- :MEASure:FREQuency?
- :MEASure:RISetime?
- :MEASure:FALLtime?
- :MEASure:PERiod?
- :MEASure:PWIDth?
- :MEASure:NWIDth?
- :MEASure:PDUTycycle?
- :MEASure:NDUTycycle?
- :MEASure:PDELay?
- :MEASure:NDELay?
- :MEASure:PPHAse?
- :MEASure:NPHAse?
- :MEASure:TOTal
- :MEASure:SOURce
- :MEASure:DELAYSOURce
- :MEASure:PHAseSOURce
- :MEASure:ENABle
- :MEASure:DISable
- :MEASure?

1. :MEASure:CLEar

Command Format:

:MEASure:CLEar

Function:

The command clears the current measurement parameters.

2. :MEASure:VPP?

Command Format:

:MEASure:VPP?□[<source>]

Function:

The command measures the Peak-Peak value of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

NOTE: "<>" in <source> indicates the source is the paramter that must be set in the commond; and "[]" indicates the parameter can be set or not set according to your demand. The followings are the same, don't repeat. Please refer to chapter 1 **Symbol Instruction** about the related explanations.

Return Format:

The query returns 5.280e000, and the unit is V.

3. :MEASure:VMAX?

Command Format:

:MEASure:VMAX?□[<source>]

Function:

The command measures the maximum of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Return Format:

The query returns 2.640e000, and the unit is V.

4. :MEASure:VMIN?

Command Format:

:MEASure:VMIN?□[<source>]

Function:

The command measures the minimum of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Return Format:

The query returns -2.640e000, and the unit is V.

5. :MEASure:VAMPlitude?

Command Format:

:MEASure:VAMPlitude?□[<source>]

Function:

The command measures the amplitude of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Return Format:

The query returns 5.280e000, and the unit is V.

6. :MEASure:VTOP?

Command Format:

:MEASure:VTOP?□[<source>]

Function:

The command measures the top value of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Reutrn Format:

The query returns 2.640e000, and the unit is V.

7. :MEASure:VBASe?

Command Format:

:MEASure:VBASe?□[<source>]

Function:

The command measures the base value of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Return Format:

The query returns -2.640e000, and the unit is V.

8. :MEASure:VAVerage?

Command Format:

:MEASure:VAVerage?□[<source>]

Function:

The command measures the average value of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Return Format:

The query returns -4.200e-003, and the unit is V.

9. :MEASure:VRMS?

Command Format:

:MEASure:VRMS?□[<source>]

Function:

The command measures the root mean square of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Return Format:

The query returns 2.460e000, and the unit is V.

10. :MEASure:OVERshoot?

Command Format:

:MEASure:OVERshoot?□[<source>]

Function:

The command measures the overshoot value of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Return Format:

The query returns 8.000e003, and the unit is V.

11. :MEASure:PREShoot?

Command Format:

:MEASure:PREShoot?□[<source>]

Function:

The command measures the preshoot value of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Return Format:

The query returns 8.000e-003, and the unit is V.

12. :MEASure:FREQuency?

Command Format:

:MEASure:FREQuency?□[<source>]

Function:

The command measures the frequency of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Return Format:

The query returns 1.000e003, and the unit is Hz.

13. :MEASure:RISetime?

Command Format:

:MEASure:RISetime?□[<source>]

Function：

The command measures the rise time of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Return Format:

The query returns 4.000e-005, and the unit is s.

14. :MEASure:FALLtime?

Command Format:

:MEASure:FALLtime?□[<source>]

Function:

The command measures the fall time of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Return Format:

The query returns 4.000e-005, and the unit is s.

15. :MEASure:PERiod?

Command Format:

:MEASure:PERiod?□[<source>]

Function:

The command measures the period of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Return Format:

The query returns 1.000e-003, and the unit is s.

16. :MEASure:PWIDth?

Command Format:

:MEASure:PWIDth?□[<source>]

Function:

The command measures the positive pulse width of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Return Format:

The query returns 5.000e-004, and the unit is s.

17. :MEASure:NWIDth?

Command Format:

:MEASure:NWIDth?□[<source>]

Function:

The command measures the negative pulse width of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Return Format:

The query returns 5.000e-004, and the unit is s.

18. :MEASure:PDUTycycle?

Command Format:

:MEASure:PDUTycycle?□[<source>]

Function:

The command measures the positive duty cycle of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Return Format:

The query returns 5.000e001, and the unit is %.

19. :MEASure:NDUTycycle?

Command Format:

:MEASure:NDUTycycle?□[<source>]

Function:

The command measures the negative duty cycle of signal from <source>. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Return Format:

The query returns 5.000e001, and the unit is %.

20. :MEASure:PDELay?

Command Format:

:MEASure:PDELay?□[<source A>,<source B>]

Function:

The command measures the delay between <sourceA> and <sourceB> relative to the rising edge. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Return Format:

The query returns <-1.000 e-004, and the unit is s.

21. :MEASure:NDELay?

Command Format:

:MEASure:NDELay?□[<source A>,<source B>]

Function:

The command measures the delay between <sourceA> and <sourceB> relative to the falling edge. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Return Format:

The query returns <-1.000 e-004, and the unit is s.

22. :MEASure:PPHAse?

Command Format:

:MEASure:PPHAse?□[<source A>,<source B>]

Function:

The command measures the phase difference between <sourceA> and <sourceB> relative to the rising edge. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Return Format:

The query returns <-1.000 e-004, and the unit is s.

23. :MEASure:NPHAse?

Command Format:

:MEASure:NPHAse?□[<source A>,<source B>]

Function:

The command measures the phase difference between <sourceA> and <sourceB> relative to the falling edge. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Return Format:

The query returns <-1.000 e-004, and the unit is s.

24. :MEASure:TOTal

Command Format:

:MEASure:TOTal□{{1|ON}|{0|OFF}}

Function:

The command sets all the measurement functions on or off.

Query Format:

:MEASure:TOTal?

Return Format:

The query returns 1 or 0, respectively indicates ON or OFF.

Example:

:MEASure:TOTal□ON Set the total measurement function on.

:MEASure:TOTal? Return 1.

25. :MEASure:SOURce

Command Format:

:MEASure:SOURce□<source>

Functions:

The command selects the measurement channel. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Query Format:

:MEASure:SOURce?

Return Format:

The query returns CH1 or CH2, CH3, CH4.

Example:

:MEASure:SOURce□CHANnel1 Measure the signal from CH1.

:MEASure:SOURce? Return CH1.

26. :MEASure:DELAYSOURce

Command Format:
:MEASure:DELAYSOURce□<source>,<source>

Functions:
The command selects the channel for measuring the time delay. The< source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Query Format:
:MEASure:DELAYSOURce?

Return Format:
The query returns CH1, CH2 or CH1, CH3, CH1, CH4, CH2, CH3, CH2, CH4, CH3, CH4.

Example:
:MEASure:DELAYSOURce□CHANnel1□CHANnel2
Seleclt measuring the time delay between CH1 and CH2.
:MEASure:DELAYSOURce?
Return CH1, CH2.

27. :MEASure:PHAseSOURce

Command Format:
:MEASure:PHAseSOURce□<source>,<source>

Functions:
The comand selects the channel for measuring the phase delay.The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Query Format:
:MEASure:PHAseSOURce?

Return Format:
The query returns CH1, CH2 or CH1, CH3, CH1, CH4, CH2, CH3, CH2, CH4, CH3, CH4.

Example:
:MEASure:PHAseSOURce□CHANnel1□CHANnel2
Measure the phase delay between CH1 and CH2.
:MEASure:PHAseSOURce? Return CH1, CH2.

28. :MEASure:ENABle

Command Format:

:MEASure:ENAble

Functions:

This command is open the locked MEASURE button and allow user to open Auto Measure.

NOTE: Only corresponding unlock command can open AUTO and Auto Measure once they are locked, neither restart nor update is available.

29. :MEASure:DISable

Command Format:

:MEASure:DISable

Function:

This command is lock MEASURE button and forbid user opening Auto Measure.

30. :MEASure?

Command Format:

:MEASure?

Function:

This command is query the state of keylock.

Acquire Commands

Acquire Commands are used to set the acquisition mode for oscilloscope.

Acquire Commands include:

- :ACQuire:TYPE
- :ACQuire:MODE
- :ACQuire:AVERages
- :ACQuire:SRATe?

1. :ACQuire:TYPE

Command Format:

:ACQuire:TYPE□<type>

Function:

The command sets the acquisition type. The <type>may be NORMal(common sample), AVERage(average sample) or PEAKdetect(peak detection).

Query Format:

:ACQuire:TYPE?

Return Format:

The query returns Normal or AVERAGE, PEAKDETECT.

Example:

:ACQuire:TYPE□AVERage Set the acquisition type as average acquisition.
:ACQuire:TYPE? Return AVERAGE.

2. :ACQuire:MODE

Command Format:

:ACQuire:MODE□<mode>

Function:

The command sets the acquisition mode. The <mode> may be RTIME (real time sample) or ETIME (equal time sample).

Query Format:

:ACQuire:MODE?

Return Format:

The query returns RTIME or ETIME.

Example:

:ACQuire:MODE□ETIME Set the acquisition mode as equal time acquisition.
:ACQuire:MODE? Return ETIME.

3. :ACQuire:AVERages

Command Format:

:ACQuire:AVERages□<count>

Function:

The command sets the average acquisition time. The <count> range is 2~256, and the count increases by the multiples of 2.

Query Format:

:ACQuire:AVERages?

Return Format:

The query returns 2 or 4, 8, 16, 32, 64, 128, 256.

Example:

:ACQuire:AVERages□16 Set the average acquisition time as 16.
:ACQuire:AVERages? Return 16.

4. :ACQuire:SRATe?

Command Format:

:ACQuire:SRATe? [{<CHANnel<n>}]

Function：

To query sample rate of CHANnel <n>, <n> may be 1、2、3、4.

Return Format:

The query returns 5.000e+05, the unit is Sa/s.

Display Commands

Diaplay Commands are used to set the display system.

Display Commands include:

- :DISPlay:TYPE
- :DISPlay:GRID
- :DISPlay:PERSist
- :DISPlay:MNUDisplay
- :DISPlay:MNUStatus
- :DISPlay:SCReen
- :DISPlay:CLEar
- :DISPlay:BRIGhtness
- :DISPlay:INTensity
- :DISPlay:DATA?

1. :DISPlay:TYPE

Command Format:

:DISPlay:TYPE□<type>

Function:

The command sets the display type of acquisition points. The <type> may be VECTors (acquisition points are connected by lines) or DOTS (acquisition points are displayed by dots).

Query Format:

:DISPlay:TYPE?

Return Format:

The query returns VECTORS or DOTS.

Example:

:DISPlay:TYPE□VECTors Set the display type as vectors.
:DISPlay:TYPE? Return VECTORS.

2. :DISPlay:GRID

Command Format:

:DISPlay:GRID□<grid>

Function:

The command sets the display type of screen grid. The <grid> may be FULL (grid and coordinate are shown), HALF (grid is not shown) or OFF (grid and coordinate are not shown).

Query Format:

:DISPlay:GRID?

Return Format:

FULL or HALF, NONE.

Example:

:DISPlay:GRID□FULL Make grid and coordinate shown.
:DISPlay:GRID? Return FULL.

3. :DISPlay:PERSist

Command Format:

:DISPlay:PERSist□{{1|ON }|{0|OFF}}

Function:

The command sets waveform persist function ON (The waveform is shown until waveform persist function is off or relevant settings are changed.) or OFF (The waveform is updated as high refresh rate).

Query Format:

:DISPlay:PERSist?

Return Format:

The query returns 1 or 0, respectively indicates ON or OFF.

Example:

:DISPlay:PERSist□ON Set waveform persist function on.
:DISPlay:PERSist? Return 1.

4. :DISPlay:MNUDisplay

Command Format:

:DISPlay:MNUDisplay□<time>

Function:

The command sets the display time of menu. The menu will hide after the display time. The <time> may be 1s, 2s, 5s, 10s, 20s or INFinite (display all the time).

Query Format:

:DISPlay:MNUDisplay?

Return Format:

The query returns 1s or 2s, 5s, 10s, 20s, Infinite.

Example:

:DISPlay:MNUDisplay□10s Set the display time as 10s.
:DISPlay:MNUDisplay? Return 10s.

5. :DISPlay:MNUStatus

Command Format:

:DISPlay:MNUStatus□{{1|ON}|{0| OFF}}

Function:

The command sets menu display function ON (Performing menu operation) or OFF (viewing the waveform).

Query Format:

:DISPlay:MNUStaus?

Return Format:

The query returns 1 or 0, respectively indicates ON or OFF.

Example:

:DISPlay:MNUStatus□ON Set menu display function on.

:DISPlay:MNUStaus? Return 1.

6. :DISPlay:SCReen

Command Format:

:DISPlay:SCReen□<scr>

Function:

The command sets the display mode of screen. The <scr>may be Normal （normal display mode） or Inverted(inverted display mode).

Query Format:

:DISPlay:SCReen?

Return Format:

The query returns NORMAL or INVERTED.

Example:

:DISPlay:SCReen□NORMal Set the screen as normal display mode.

:DISPlay:SCReen? Return NORMAL.

7. :DISPlay:CLEar

Command Format:

:DISPlay:CLEar

Function:
The command clears the last waveform on the screen.

8. :DISPlay:BRIGhtness

Command Format:
:DISPlay:BRIGhtness□<count>

Function:
The command sets the brightness of grid. The <count> range is 0~100, and the bigger the count is, the brighter the grid becomes.

Query Format:
:DISPlay:BRIGhtness?

Return Format:
The query returns 0 or 1, 2 ……100 .

Example:
:DISPlay:BRIGhtness□10 Set the grid bightness as 10.
:DISPlay:BRIGhtness? Return 10.

9. :DISPlay:INTensity

Command Format:
:DISPlay:INTensity□<count>

Function:
The command sets the brightness of the waveform. The <count> range is 0~100, and the bigger the count is, the brighter the waveform becomes.

Query Format:
:DISPlay:INTensity?

Return Format:
The query returns 0 or 1, 2 ……100.

Example:
:DISPlay:INTensity□12 Set the waveform brightness as 12.
:DISPlay:INTensity? Return 12.

10. :DISPlay:DATA?

Command Format:

:DISPlay:DATA?

Function:

The command queries image data on the current screen, and the data format accords with IEEE488.2 standard. The data structure is: #800078788+the data of 8 bit bitmap.

Vertical Commands

Vertical Commands are used to set the vertical system from every channel separately.

Vertical Commands include:

- :CHANnel<n>:BWLimit
- :CHANnel<n>:COUPling
- :CHANnel<n>:DISPlay
- :CHANnel<n>:INVert
- :CHANnel<n>:OFFSet
- :CHANnel<n>:PROBe
- :CHANnel<n>:SCALe
- :CHANnel<n>:FILTer
- :CHANnel<n>:MEMoryDepth?
- :CHANnel<n>:UNITs
- :CHANnel<n>:VERNier

1. :CHANnel<n>:BWLimit

Command Format:

:CHANnel<n>:BWLimit□{{1|ON}|{0|OFF}}

Function:

The command sets bandwidth limit function ON (limit band width to 20MHz to reduce noise) or OFF (full band width). The <n> may be 1, 2, 3 or 4.

Query Format:

:CHANnel<n>:BWLimit?

Return Format:

The query returns 1 or 0, respectively indicates ON or OFF.

Example:

:CHANnel2:BWLimi□OFF Set BW limit function of channel 2 off.
:CHANnel2:BWLimit? Return 0.

2. :CHANnel<n>:COUPling

Command Format:

:CHANnel<n>:COUPling□{DC | AC | GND}

Function:

The command sets coupling mode as DC (both AC and DC components of the input signal can pass), AC (the DC component of the input signal can not pass) or GND (disconnect the input signal). The <n> may be 1, 2, 3 or 4.

Query Format:

:CHANnel<n>:COUPling?

Return Format:

The query returns AC or DC, GND.

Example:

:CHANnel2:COUPling□DC Set the coupling mode of channel 2 as DC.
:CHANnel2:COUPling? Return DC.

3. :CHANnel<n>:DISPlay

Command Format:

:CHANnel<n>:DISPlay□{{1|ON}|{0|OFF}}

Function:

The command sets the channel ON or OFF. The <n> may be 1, 2, 3 or 4.

Query Format:

:CHANnel<n>:DISPlay?

Return Format:

The query returns 1 or 0, respectively indicates ON or OFF.

Examples:

:CHANnel2:DISPlay□ON Set channel 2 on.
:CHANnel2:DISPlay? Return 1.

4. :CHANnel<n>:INVert

Command Format:

:CHANnel<n>:INVert□{{1|ON}|{0|OFF}}

Function:

The command sets waveform invert function ON (the inverted waveform is display) or OFF (the normal waveform is display). The <n> may be 1, 2, 3 or 4.

Query Format:

:CHANnel<n>:INVert?

Return Format:

The query returns 1 or 0, respectively indicates ON or OFF.

Example:

:CHANnel2:INVert□OFF Set the invert function of channel 2 off.
:CHANnel2:INVert? Return 0.

5. :CHANnel<n>:OFFSet

Command Format:

:CHANnel<n>:OFFSet□<offset>

Function:

The command sets the vertical offset. The <n> may be 1, 2, 3 or 4.
Scale >200mV, <offset>: -40V~ +40V;
Scale ≤200mV, <offset>: -2V ~ +2V.

Query Format:

:CHANnel<n>:OFFSet?

Return Format:

The query returns the value of offset, and the unit is V.

Example:

:CHANnel2:OFFSet□20 Set the vertical offset of channel 2 as 20V.
:CHANnel2:OFFSet? Return 2.000e+01.

6. :CHANnel<n>:PROBe

Command Format:

:CHANnel<n>:PROBe□<attn>

Function:

The command sets the attenuation factor of probe. The <n> may be 1, 2, 3 or 4, and the <attn> may be 0.001 X, 0.01 X, 0.1 X, 1X, 2 X, 5X, 10X, 20 X, 50X, 100X, 200 X, 500X or 1000X.

Query Format:

:CHANnel<n>:PROBe?

Return Format:

The query returns the value of attenuation factor.

Example:

:CHANnel2:PROBe□10X Set the annenuation factor of the probe of channel2 as 10X.
:CHANnel2:PROBe? Return 10X.

7. :CHANnel<n>:SCALe

Command Format:

:CHANnel<n>:SCALe□<range>

Function:

The command sets the vertical scale for magnifying waveform. The <n> may be 1, 2, 3 or 4.

Probe 0.001X, <range>: 2μV ~ 10mV;
Probe 0.01X, <range>: 20μV ~ 100mV;
Probe 0.1X, <range>: 200μV ~ 1V;
Probe 1X, <range>: 2mV ~ 10V;
Probe 2X, <range>: 4mV ~ 20V;
Probe 5X, <range>: 10mV ~ 50V;
Probe 10X, <range>: 20mV ~ 100V;
Probe 20X, <range>: 40mV ~ 200V;
Probe 50X, <range>: 100mV ~ 500V;
Probe 100X, <range>: 200mV ~ 1kV;
Probe 200X, <range>: 400mV ~ 2kV;
Probe 500X,<range>: 1V ~ 5kV;
Probe 1000X, <range>: 2V ~ 10kV.

Query Format:

:CHANnel<n>:SCALe?

Return Format:

The query returns the value of vertical scale, and the unit is V.

Example:

:CHANnel2:PROBe□10X Set the attenuation factor of channel 2 as 10X.
:CHANnel2:SCALe□20 Set the vertical scale of channel 2 as 20V.
:CHANnel2:SCALe? Return 2.000e+01.

8. :CHANnel<n>:FILTer

Command Format:

:CHANnel<n>:FILTer□{{1|ON}|{0|OFF}}

Function:

The command sets digital filter function ON or OFF. The <n> may be 1, 2, 3 or 4.

Query Format:

:CHANnel<n>:FILTer?

Return Format:

The query returns 1 or 0, respectively indicates ON or OFF.

Example:

:CHANnel2:FILTer□OFF Set the digital filter of channel 2 off.

:CHANnel2:FILTer? Return 0.

9. :CHANnel<n>:MEMoryDepth?

Command Format:

:CHANnel<n>:MEMoryDepth?

Function:

This command is query the memory depth on channel x

There are three instances:

1).Alternate trigger: 8192

2).Slow scan or ROLL: 0~8192

3).Others: 8192

NOTE: In Slow scan mode: In the event of the horizontal timebase is set as 50ms/div or more slowly, the instrument will turn into Slow scan mode. Under this circumstance, the oscilloscope will gather the datum form the left side of the trigger point and then continue gathering the waves form the right side after triggering. If use Slow scan mode to observe the low frequency signal, you are suggested to set the couple mode of channel as DC .

10. :CHANnel<n>:UNITs

Command Format:

:CHANnel<n>:UNITs <units>

Function:

The command sets the unit as VOLTs (V), AMPeres (A), WATTs (W) or UNKNown. The <n> may be 1, 2, 3 or 4.

Query Format:

:CHANnel<n>:UNITs?

Return Format:

The query returns VOLTs or AMPeres, WATTs, UNKNown.

Example:

:CHANnel1:UNITs VOLTs Set the unit of channel 1 as V.

:CHANnel1:UNITs? Return VOLTs.

11. :CHANnel<n>:VERNier

Command Format:

:CHANnel<n>:VERNier□{{1|ON}|{0|OFF}}

Function:

The command sets the adjustment mode of vertical scale as ON (Fine) or OFF (Coarse). The vertical scale steps by 1-2-5 in Coarse mode; and by equality in Fine mode.The <n> may be 1, 2, 3, 4.

Query Format:

:CHANnel<n>:VERNier?

Return Format:

The query returns 1 or 0, respectively indicates ON or OFF.

Example:

:CHANnel2:VERNier□ON Set the fine adjustment function of channel 2 on.

:CHANnel2:VERNier? Return 1.

Horizontal Commands

Horizontal Commands are used to set horizontal scale and horizontal offset. Changing horizontal scale makes the waveform enlarge or shrink; and changing horizontal position will lead the waveform offset relative to center screen.

Horizontal Commands include:

- :TIMebase:MODE
- :TIMebase:DELayed:OFFSet
- :TIMebase:DELayed:SCALe
- :TIMebase[:MAIN]:OFFSet
- :TIMebase[:MAIN]:SCALe
- :TIMebase:FORMat

1. :TIMebase:MODE

Command Format:

:TIMebase:MODE□<mode>

Function:

The command sets the scan mode of horizontal timebase as MAIN (main time base) or DELayed (delayed scan time base).

Query Format:

:TIMebase:MODE?

Return Format:

The query returns MAIN or DELAYED.

Example:

:TIMebase:MODE□MAIN Set the scan mode as main time base.
:TIMebase:MODE? Return MAIN.

2. :TIMebase:DELayed:OFFSet

Command Format:

:TIMebase:DELayed:OFFSet□<offset>

Function:

The command sets the timebase offset of delayed scan, that is the offset of the waveform position relative to center screen.
In NORMAL mode, <offset>: 1s ~ memory capacitance;
In STOP mode, <offset>: -500s ~ +500s;
In SCAN mode,<offset>:
(-6 *MainScale + 6 DelayedScale①) ~(6 *MainScale - 6 DelayedScale)
Scale is the current horizontal scale, and the unit is s/div.

NOTE①: Delay, the time range: +/- 6*MainScale; The length of time is 12*DelayedScale;
So Delayed Offset range is:
(-6MainScale + 6*DelayedScale)~(6*MainScale-6*DelayedScale).
For example: When Main 5ms，Delay 2ms, EMS memory time is +/- 6*5=30ms,
Delay time is 6*2 = 12ms. Delay Offset range: (-30+6) ~ (30-6) ms.

Query Format:

:TIMebase:DELayed:OFFSet?

Return Format:

The query reutrns the value of offset, and the unit is s.

Example:

:TIMebase:MODE□DELayed Set the scan mode as delayed scan.

:TIMebase:DELayed:OFFSet□1 Set the timebase offset as 1s.
:TIMebase:DELayed:OFFSet? Return 1.000e+00.

3. :TIMebase:DELayed:SCALe

Command Format:

:TIMebase:DELayed:SCALe□<scale_val>

Function:

The command sets the timebase scale of delayed scan, and the unit is s/div. When the "Delayed" is "ON", for view waveform details, the waveform may be amplified under the width of window vary with the delayed timebase scale.

- In NoRMAL mode, different types of instruments have different sweep ranges:
  DS1204B instrument，<scale_val> range：1ns/div~50s/div.
  DS1104B instrument，<scale_val>range：2ns/div~50s/div.
  DS1064B instrument，<scale_val>range：5ns/div~50s/div.
- In SCAN mode, <scale_val>range: 50ms ~ 50s.

Query Format:

:TIMebase:DELayed:SCALe?

Return Format:

The query returns the value of timebase scale, and the unit is s.

Example:

:TIMebase:MODE□DELayed Set the scan mode as delayed scan.
:TIMebase:DELayed:SCALe□2 Set the timebase scale as 2s.
:TIMebase:DELayed:SCALe? Return 2.000e+00.

4. :TIMebase[:MAIN]:OFFSet

Command Format:

:TIMebase[:MAIN]:OFFSet□<offset>

Function:

The command sets the timebase offset of main mode, that is the offset of the waveform position relative to center screen.

In NORMAL mode, <offset>: 1s ~ memory capacitance;

In STOP mode, <offset>: -500s ~ +500s;

In SCAN mode,<offset>:

(-6 *MainScale + 6 DelayedScale①) ~(6 *MainScale - 6 DelayedScale)

Scale is the current horizontal scale, and the unit is s/div.

NOTE①: Delay, the time range: +/- 6*MainScale; The length of time is 12*DelayedScale;

So Delayed Offset range is:

(-6MainScale + 6*DelayedScale)~(6*MainScale-6*DelayedScale).

For example: When Main 5ms，Delay 2ms, EMS memory time is +/- 6*5=30ms,

Delay time is 6*2 = 12ms. Delay Offset range: (-30+6) ~ (30-6) ms.

In MAIN mode, [:MAIN] may be omitted.

Query Format:

:TIMebase:MAIN:OFFSet?

Return Format:

The query returns the value of timebase offset, and the unit is s.

Example:

:TIMebase:MODE□MAIN Set the scan mode as main.

:TIMebase:OFFSet□1 Set the timebase offset as 1s.

:TIMebase:OFFSet? Return 1.000e+00.

5. :TIMebase[:MAIN]:SCALe

Command Format:

:TIMebase[:MAIN]:SCALe□<scale_val>

Function:

The command sets the timebase scale of main mode, and the unit is s/div.

- In NoRMAL mode, different types of instruments have different sweep ranges:
  DS1204B instrument，<scale_val> range：1ns/div~50s/div.
  DS1104B instrument，<scale_val>range：2ns/div~50s/div.
  DS1064B instrument，<scale_val>range：5ns/div~50s/div.
- In SCAN mode, <scale_val>range: 50ms ~ 50s.

Query Format:
:TIMebase[:MAIN]:SCALe?

Return Format:
The query returns the value of timebase scale, and the unit is s.

Example:
:TIMebase:MODE□MAIN Set the scan mode as main.
:TIMebase:SCALe□2 Set the timebase scale as 2s.
:TIMebase:SCALe? Return 2.000e+00.

6. :TIMebase:FORMat

Command Format:
:TIMebase:FORMat□<vlaue>

Function:
The command sets the timebase format as XY (the amplitude of channel 1 is shown in Y axis, and the amplitude of channel 2 is shown in Y axis), YT (the relationship between the voltage and the time is shown) or ROLL (the acquisition points on screen are updated from left to right).

Query Format:
:TIMebase:FORMat?

Return Format:
The query returns X-Y or Y-T, ROLL.

Example:
:TIMebase:FORMat□YT Set the timebase format as Y-T.
:TIMebase:FORMat? Return Y-T.

Trigger Commands

Trigger Control Commands

Trigger system makes the meaningful waveform shown steadily. Trigger determines when the oscilloscope starts to acquire data and to display a waveform. When trigger is set up properly, it can convert unstable displays into meaningful waveforms.

When the oscilloscope starts to acquire data, firstly enough data are needed to be collected so as to shape into a waveform on the left of the trigger point. The oscilloscope continues acquiring data while waiting for the trigger condition to occur. After it detects a trigger, the oscilloscope continues to acquire enough data so that it can display the waveform on the right of the trigger point.

Trigger Control Commands include:

- :TRIGger:MODE
- :TRIGger<mode>:SOURce
- :TRIGger<mode>:LEVel
- :TRIGger<mode>:SWEep
- :TRIGger:COUPling
- :TRIGger:HOLDoff
- :TRIGger:STATus
- :TRIGger:SENSitivity
- :TRIGger:HFREject

1. :TRIGger:MODE

Command Format:

:TRIGger:MODE□<mode>

Function:

The command sets the trigger mode as Edge, Pulse, Video, Alternation or Pattern trigger.

Query Format:

:TRIGger:MODE?

Return Format:

The query returns EDGE or PULSE, VIDEO, ALTERNATION, PATTERN.

Example:

:TRIGger:MODE□EDGE Set the trigger mode as edge trigger.
:TRIGger:MODE? Return EDGE.

2. :TRIGger<mode>:SOURce

Command Format:

:TRIGger<mode>:SOURce□<source>

Function:

The command sets the trigger source as channel (CH1, CH2, CH3, CH4), external trigger (EXT, EXT5) or AC Line.
The< mode> is :EDGE, and the <source> may be CHANnel<n>, EXT, EXT5 or ACLine;
The< mode> is :PULSE, and the <source> may be CHANnel<n>, EXT or EXT5;
The< mode> is :VIDEO, and the <source> may be CHANnel<n>, EXT or EXT5;
The <mode> is :PATTern, <source> may be CHANnel<n>, EXT.
The <n> may be 1, 2, 3 or 4.

Query Format:

:TRIGger<mode>:SOURce?

Return Format:

The query returns CH1 or CH2, CH3, CH4, EXT, EXT5, ACLINE.

Example:

:TRIGGER:EDGE:SOURce□CHANnel1 Set the edge trigger source as channel 1.

:TRIGGER:EDGE:SOURce? Return CH1.

3. :TRIGger<mode>:LEVel

Command Format:

:TRIGger<mode>:LEVel□<level>

Function:

The command sets the voltage level of Edge, Pulse or Video trigger. The <mode> may be :EDGE, :PULSe or :VIDEO or PATTern.

<level> range: (- 6*Scale-Offset①) ~(+6*Scale+Offset①).

Scale is the current vertical scale, and the unit is V/div. If choose PATTern mode, it needs SOURce parameter.

NOTE①: TtigLevel range is up to+/-6 Scale, when channel has offset, it needs to detract offset ,such as 1V tap positon, 1V offset, the trigger range is -7V~5V.

Query Format:

:TRIGger<mode>:LEVel?

Return Format:

The query returns the value of voltage level, and the unit is V.

Example:

:TRIGger:EDGE:LEVel□2 Set the trigger level as 2V.

:TRIGger:EDGE:LEVel? Return 2.000e+000.

4. :TRIGger<mode>:SWEep

Command Format:

:TRIGger<mode>:SWEep□{ AUTO | NORMal |SINGle }

Function:

The command sets the trigger mode.

AUTO (When no trigger is existing, the system will generate a trigger signal to force trigger.);

NORMal (Acquire waveform when trigger occurred);

SINGle (Execute once trigger when all the condition are marched and stop).

The <mode> may be :EDGE, :PULSe or :PATTern.

Query Format:
:TRIGger<mode>:SWEep?

Return Format:
The query returns AUTO or NORMAL, SINGLE.

Example:
:TRIGger:EDGE :SWEep□AUTO Set the trigger type as AUTO.
:TRIGger:EDGE :SWEep? Return AUTO.

5. :TRIGger:COUPling

Command Format:
:TRIGger:COUPling□{ DC | AC | LF }

Function:
The command sets the coupling mode.
DC (Allow all signals pass);
AC (Reject DC signals and attenuate the signal below 10Hz);
LF (Reject DC signals and attenuate the signals below 8kHz).

Query Format:
:TRIGger:COUPling?

Return Format:
The query returns DC or AC, LF.

Example:
:TRIGger:COUPling□DC Set the coupling mode as DC.
:TRIGger:COUPling? Return DC.

6. :TRIGger:HOLDoff

Command Format:
:TRIGger:HOLDoff□<count>

Function:
The command sets the holdoff time of a trigger. Holdoff time indicates the waiting time before oscilloscope starts a new trigger. During Holdoff,

oscilloscope will not trigger until Holdoff ends.
The <count> range is 100ns~1.5s.

Query Format:
:TRIGger:HOLDoff?

Return Format:
The query returns the value of holdoff time, and the unit is s.

Example:
:TRIGger:HOLDoff□0.0001 Set the holdoff time as 100μs.
:TRIGger:HOLDoff? Return 1.000e-04.

7. :TRIGger:STATus

Query Format:
:TRIGger:STATus?

Function:
The command queries the current status of the oscilloscope. The status may be RUN, STOP, T' D, WAIT,SCAN or AUTO.

Return Format:
The query returns RUN or STOP, T'D, WAIT, AUTO.

8. :TRIGger:SENSitivity

Command Format:
:TRIGger:SENSitivity□<count>

Function:
The command sets the trigger sensitivity. The <count> range is 0.1div~1div.

Query Format:
:TRIGger:SENSitivity?

Return Format:
The query returns the value of trigger sensitivity, and the unit is div.

Example:

:TRIGger:SENSitivity□0.2 Set the trigger sensitivity as 0.2div.

:TRIGger:SENSitivity? Return 2.00e-01.

9. :TRIGger:HFREject

Command Format:

:TRIGger:HFREject□{{1|ON}|{0|OFF}}

Function:

The command sets high frequency reject function on or off.

Query Format:

:TRIGger:HFREject?

Return Format:

The query returns 1 or 0, respectively indicates ON or OFF.

Example:

:TRIGger:HFREject□ON Set HFR on.

:TRIGger:HFREject? Return 1.

Trigger Mode Commands

Trgger Mode includes: Edge, Pulse, Video, Alternation, and Pattern trigger.

Trigger Mode Commands include:

- :TRIGger:EDGE:SLOPe
- :TRIGger:PULSe:MODE
- :TRIGger:PULSe:WIDTh
- :TRIGger:VIDEO:MODE
- :TRIGger:VIDEO:POLarity
- :TRIGger:VIDEO:STANdard
- :TRIGger:VIDEO:LINE
- :TRIGger:ALTernation:SOURce
- :TRIGger:ALTernation:CURRentSOURce
- :TRIGger:ALTernation:TYPE
- :TRIGger:ALTernation:TimeSCALe
- :TRIGger:ALTernation:TimeOFFSet
- :TRIGger:ALTernation:LEVel
- :TRIGger:ALTernation:EDGE:SLOPe
- :TRIGger:ALTernation:PULSe:MODE
- :TRIGger:ALTernation:PULSe:TIME
- :TRIGger:ALTernation:VIDEO:POLarity
- :TRIGger:ALTernation:VIDEO:STANdard
- :TRIGger:ALTernation:VIDEO:MODE
- :TRIGger:ALTernation:HOLDoff
- :TRIGger:ALTernation:VIDEO:LINE
- :TRIGger:ALTernation:COUPling
- :TRIGger:ALTernation:SENSitivity
- :TRIGger:PATTern:PATTern

1. Edge Trigger

Edge trigger occurs when the trigger input meets a specified level along the specified slope direction.

(1) :TRIGger:EDGE:SLOPe

Command Format:

:TRIGger:EDGE:SLOPe□{ POSitive | NEGative | ALTernation }

Function:

The command sets the trigger edge as Positive (rising edge), Negative (falling edge) or ALTernation (rising and falling edge).

Query Format:

:TRIGger:EDGE:SLOPe?

Return Format:

The query returns POSITIVE or NEGATIVE, ALTERNATION.

Example:

:TRIGger:EDGE:SLOPe□POSitive	Set the trigger edge as rise edge.
:TRIGger:EDGE:SLOPe?	Return POSITIVE.

2. Pulse Trigger

Setup a certain trigger condition to capture specified pulse. Pulse tigger occurs when pulse width meets the condition.

(1) :TRIGger:PULSe:MODE

Command Format:

:TRIGger:PULSe:MODE□<mod>

Function:

The command sets the trigger condition as +Greaterthan (positive pulse width greater than), +Lessthan (positive pulse width less than), + Equal (positive pulse width equal), -Greaterthan (negative pulse width greater than), -Lessthan (negative pulse width less than) or –Equal (negative pulse width equal).

Query Format:

:TRIGger:PULSe:MODE?

Return Format:

The query returns +GREATER THAN or +LESS THAN, +EQUAL, -GREATER THAN, -LESS THAN, –EQUAL.

Example:

:TRIGger:PULSe:MODE□+GREATERTHAN

Set the trigger condition as positive pulse widrh greater than.

:TRIGger:PULSe:MODE? Return +GREATER THAN.

(2) :TRIGger:PULSe:WIDTh

Command Format:

:TRIGger:PULSe:WIDTh□<wid>

Function:

The command sets the pulse width. The <wid> range is 20ns ~10s.

Query Format:

:TRIGger:PULSe:WIDTh?

Return Format:

The query returns the value of pulse width, and the unit is s.

Example:

:TRIGger:PULSe:WIDTh□0.001 Set the pulse width as 1ms.

:TRIGger:PULSe:WIDTh? Return 1.000e-03.

3. Video Trigger

In video trigger, the standard video signal (NTSC, PAL/SECAM) is triggered in field or line. The trigger coupling mode is DC.

(1) :TRIGger:VIDEO:MODE

Command Format:

:TRIGger:VIDEO:MODE□<mode>

Function:

The command sets the trigger sync mode as ODDFIELD, EVENFIELD, LINE or ALLLINES.

Query Format:

:TRIGger:VIDEO:MODE?

Return Format:

The query returns ODD FIELD ro EVEN FIELD, LINE, ALL LINES.

Example:

:TRIGger:VIDEO:MODE□EVENFIELD

Set the trigger sync mode as even field.

:TRIGger:VIDEO:MODE? Return EVEN FIELD.

(2) :TRIGger:VIDEO:POLarity

Command Format:

:TRIGger:VIDEO:POLarity□{ POSitive | NEGative }

Function:

The command sets the video polarity as POSitive (it is applicable for the video signal that the black level is low) or NEGative (the black level is high).

Query Format:

:TRIGger:VIDEO:POLarity?

Return Format:

The query returns POSITIVE or NEGATIVE.

Example:

:TRIGger:VIDEO:POLarity□POSitive Set the video polarity as positive.
:TRIGger:VIDEO:POLarity? Return POSITIVE.

(3) :TRIGger:VIDEO:STANdard

Command Format:

:TRIGger:VIDEO:STANdard□{ NTSC | PALSecam }

Function:

The command sets the video standard as NTSC or PAL/SECAM.

Query Format:

:TRIGger:VIDEO:STANdard?

Return Format:

The query returns NTSC or PAL/SECAM.

Example:

:TRIGger:VIDEO:STANdard□PALSECAM
Set the video standard as PAL/SECAM.
:TRIGger:VIDEO:STANdard? Return PAL/SECAM.

(4) :TRIGger:VIDEO:LINE

Command Format:

:TRIGger:VIDEO:LINE□<value>

Function:

The command sets the number of sync specified line. In NTSC standard, the <value> range is 1~525; in PAL standard, the <value> range is 1~625.

Query Format:

:TRIGger:VIDEO:LINE?

Return Format:

The query returns the number of specified line.

Example:

:TRIGger:VIDEO:LINE□25 Set the number of sync specified line as 25.
:TRIGger:VIDEO:LINE? Return 25.

4. Alternation Trigger

In alternation trigger, different signals can be triggered at the same time. And the trigger signals are from two vertical channels. Alternation trigger is applicable for viewing two non-relative signals. The trigger mode of two vertical channels can be set as EDGE, PULSe, or VIDEO trigger.

(1) :TRIGger:ALTernation:SOURce

Command Format:

:TRIGger:ALTernation:SOURce□<source>

Function:

The command selects the alternation trigger channel. The <source> may be CH1CH2, CH1CH3, CH1CH4, CH2CH3, CH2CH4 or CH3CH4.

Query Format:

:TRIGger:ALTernation:SOURce?

Return Format:

The query returns CH1CH2 or CH1CH3, CH1CH4, CH2CH3, CH2CH4, CH3CH4.

Example:

:TRIGger:ALTernation:SOURce□CH1CH2
Set the alternation channel as CH1CH2.
:TRIGger:ALTernation:SOURce? Return CH1CH2.

(2) :TRIGger:ALTernation:CURRentSOURce

Command Format:

:TRIGger:ALTernation:CURRentSOURce□<source>

Function:

The command sets the current channel. The <source> may be SOURceA or SOURceB, and the source A and B are varying with the current alternation channel.

Query Format:

:TRIGger:ALTernation:CURRentSOURce?

Return Format:

The query returns SOURceA or SOURceB.

Example:

:TRIGger:ALTernation:SOURce□CH1CH2

Set the alternation channel as CH1CH2.

:TRIGger:ALTernation:CURRentSOURce SOURceB

Set the current channel as source B.

:TRIGger:ALTernation:CURRentSOURce? Return SOURceB.

(3) :TRIGger:ALTernation:TYPE

Command Format:

:TRIGger:ALTernation:TYPE□<value>[,<source>]

Function:

The command sets the trigger type. The <value> may be EDGE, PULSe or VIDEO, and the <source> may be SOURceA or SOURceB, and the source A and B are varying with the current alternation channel.

Query Format:

:TRIGger:ALTernation:TYPE?□[<source>]

Return Format:

The query returns EDGE or PULSE, VIDEO.

Example:

:TRIGger:ALTernation:TYPE□EDGE,SOURCeB

Set the trigger type as edge trigger.

:TRIGger:ALTernation:TYPE?□SOURCeB Return EDGE.

(4) :TRIGger:ALTernation:TimeSCALe

Command Format:

:TRIGger:ALTernation:TimeSCALe□<value>[,<source>]

Function:

The command sets the time scale of current channel. The <value> range is 2ns~20ms, and the <source> may be SOURceA or SOURceB, and the source A and B are varying with the current alternation channel.

- In NoRMAL mode, different types of instruments have different sweep ranges:
  DS1204B instrument，<scale_val> range：1ns/div~50s/div.
  DS1104B instrument，<scale_val>range：2ns/div~50s/div.
  DS1064B instrument，<scale_val>range：5ns/div~50s/div.
- In SCAN mode, <scale_val>range: 50ms ~ 50s.

Query Format:
:TRIGger:ALTernation:TimeSCALe?□[<source>]

Return Format:
The query returns the value of time scale, and the unit is s.

Example:
:TRIGger:ALTernation:TimeSCALe□0.001,SOURCeB
Set the time scale as 1ms。
:TRIGger:ALTernation:TimeSCALe?□SOURCeB Return 1.000e-03.

(5) :TRIGger:ALTernation:TimeOFFSet

Command Format:
:TRIGger:ALTernation:TimeOFFSet□<value>[,<source>]

Function:
The command sets the timebase offset.
In NORMAL mode, <value>: 1s ~ memory capacitance;
In STOP mode, <value: -500s ~ +500s;

Query Format:
:TRIGger:ALTernation:TimeOFFSet?□[<source>]

Return Format:
The query returns the value of timebase offset, and the unit is s.

Example:
:TRIGger:ALTernation:TimeOFFSet□0.0002,SOURceB
Set the timebase offset as 200μs.
:TRIGger:ALTernation:TimeOFFSet?□SOURceB Return 2.000e-04.

(6) :TRIGger:ALTernation:LEVel

Command Format:

:TRIGger:ALTernation:LEVel□<value>[,<source>]

Function:

The command sets the trigger level of current channel. The <value> range <value> range: (- 6*Scale-Offset①) ~(+6*Scale+Offset①).Scale is the current vertical scale, and the unit is V/div. the <source> may be SOURceA or SOURceB, and the source A and B is different according to the current alternation channel.

NOTE①: TtigLevel range is up to+/-6 Scale, when channel has offset, it needs to detract offset ,such as 1V tap positon, 1V offset, the trigger range is -7V~5V.

Query Format:

:TRIGger:ALTernation:LEVel?□[<source>]

Return Format:

The query returns the value of trigger voltage level, and the unit is V.

Example:

:TRIGGER:ALTERNATION:LEVEL□2, SOURceB

Set the trigger voltage level as 2V.

:TRIGGER:ALTERNATION:LEVEL?□SOURceB Return 2.00e+00.

(7) :TRIGger:ALTernation:EDGE:SLOPe

Command Format:

:TRIGger:ALTernation:EDGE:SLOPe□<value>[,<source>]

Function:

The command sets the edge type of edge trigger in current channel as POSitive (rising edge) or NEGative (falling edge). The <source> may be SOURceA or SOURceB, and the source A and B are varying with the current alternation channel.

Query Format:

:TRIGger:ALTernation:EDGE:SLOPe?□[<source>]

Return Format:

The query returns POSITIVE or NEGATIVE.

Example:
:TRIGger:ALTernation:EDGE:SLOPe□POSitive, SOURceB
Set the edge type as rising edge.
:TRIGger:ALTernation:EDGE:SLOPe?□SOURceB Return POSITIVE.

(8) :TRIGger:ALTernation:PULSe:MODE

Command Format:
:TRIGger:ALTernation:PULSE:MODE□<value>[,<source>]

Function:
The command sets the trigger condition of pulse trigger. The <value> may be +GREaterthan, +LESSthan, + EQUal, -GREaterthan, -LESSthan or –EQUal. The <source> may be SOURceA or SOURceB, and the source A and B are varying with the current alternation channel.

Query Format:
:TRIGger:ALTernation:PULSe:MODE?□[<source>]

Return Format:
The query returns +GREATER THAN or +LESS THAN, +EQUAL, -GREATER THAN, -LESS THAN, –EQUAL.

Example:
:TRIGger:ALTernation:PULSe:MODE□+GREATERTHAN, SOURceB
Set the trigger condition as +GREATER THAN.
:TRIGger:ALTernation:PULSe:MODE?□SOURceB
Return +GREATER THAN.

(9) :TRIGger:ALTernation:PULSe:TIME

Command Format:
:TRIGger:ALTernation:PULSe:TIME□<value>[,<source>]

Function:
The command sets the pulse width, the value range is 20ns~10s. The <source> may be SOURceA or SOURceB, and the source A and B are varying with the current alternation channel.

Query Format:
:TRIGger:ALTernation:PULSe:TIME?□[<source>]

Return Format:
The query returns the value of pulse width, and the unit is s.

Example:
:TRIGGER:ALTERNATION:PULSe:TIME□0.002, SOURceB
Set the pulse width as 2ms.
:TRIGGER:ALTERNATION:PULSeE:TIME?□SOURCeB Return 2.000e-03.

(10) :TRIGger:ALTernation:VIDEO:POLarity

Command Format:
:TRIGger:ALTernation:VIDEO:POLarity□{POSitive | NEGative }[,<source>]

Function:
The command sets the video polarity as Positive or Negative. The <source> may be SOURceA or SOURceB, and the source A and B are ivarying with the current alternation channel.

Query Format:
:TRIGger:ALTernation:VIDEO:POLarity?□[<source>]

Return Format:
The query returns POSITIVE or NEGATIVE.

Example:
:TRIGger:ALTernation:VIDEO:POLarity□POSitive, SOURceB
Set the video polarity as positive.
:TRIGger:ALTernation:VIDEO:POLarity?□SOURceB Return POSITIVE.

(11) :TRIGger:ALTernation:VIDEO:STANdard

Command Format:
:TRIGger:ALTernation:VIDEO:STANdard□{NTSC | PALSecam}[,<source>]

Function:
The command sets the video standard as NTSC or PAL/SECAM. The <source> may be SOURceA or SOURceB, and the source A and B are varying with the current alternation channel.

Query Format:
:TRIGger:ALTernation:VIDEO:STANdard?□[<source>]

Return Format:
The query returns NTSC or PAL/SECAM.

Example:
:TRIGger:ALTernation:VIDEO:STANdard□NTSC, SOURceB
Set the video standard as NTSC.
:TRIGger:ALTernation:VIDEO:STANdard?□SOURceB Return NTSC.

(12) :TRIGger:ALTernation:VIDEO:MODE

Command Format:
:TRIGger:ALTernation:VIDEO:MODE□{ALLLins | ODDField | EVENfield | LINE}[,<source>]

Function:
The command sets the sync mode of alternation trigger and video trigger as ALLLines, ODDField, EVENfield or LINE. The <source> may be SOURceA or SOURceB, and the source A and B are varying with the current alternation channel.

Query Format:
:TRIGger:ALTernation:VIDEO:MODE?□[<source>]

Return Format:
The query returns ALL LINES or ODD FIELD, EVEN FIELD, LINE.

Example:
:TRIGger:ALTernation:VIDEO:MODE□ALLLines,SOURceB
Set the sync mode as all lines.
:TRIGger:ALTernation:VIDEO:MODE?□SOURceB Return ALL LINES.

(13) :TRIGger:ALTernation:HOLDoff

Command Format:

:TRIGger:ALTernation:HOLDoff□<count>[,<source>]

Function:

The command sets the holdoff time to trigger the specified source alternately. Holdoff time is the waiting time of oscilloscope before starting a new trigger. During Holdoff, oscilloscope will not trigger until Holdoff ends. The <count> range is 100ns~1.5s. The <source> may be SOURceA or SOURceB.

Query Format:

:TRIGger:ALTernation:HOLDoff?□[<source>]

Return Format:

The query returns the value of holdoff time, and the unit is s.

Example:

:TRIGger:HOLDoff□0.0001, SOURceA

Set the holdoff time of source A as 100us.

:TRIGger:HOLDoff?□SOURceA Return 1.000e-04.

(14) :TRIGger:ALTernation:VIDEO:LINE

Command Format:

:TRIGger:ALTernation:VIDEO:LINE□<value>[,<source>]

Function:

The command sets the number of sync specified line. In NTSC standard, the <value> range is 1~525; in PAL/SECAN standard, the <value> range is 1~625. The <source> may be SOURceA or SOURceB, and the source A and B are varying with the current alternation channel.

Query Format:

:TRIGger:ALTernation:VIDEO:LINE?□[<source>]

Return Format:

Thte query returns the number of specified line.

Example:

:TRIGGER:ALTERNATION:VIDEO:LINE□100, SOURceB

Set the specified line number as 100.

:TRIGger:ALTernation:VIDEO:LINE?□SOURceB Return 100.

(15) :TRIGger:ALTernation:COUPling

Command Format:

:TRIGger:ALTernation:COUPling□{DC | AC | HF | LF}[,<source>]

Function:

The command sets the coupling mode.

DC (Allow all signals pass);

AC (Reject DC signals and attenuate AC signals below 10Hz).

HF (Reject high frequency signals above 150 kHz).

LF (Reject DC and attenuate low frequency signals below 8kHz).

The <source> may be SOURceA or SOURceB, and the source A and B are varying with the current alternation channel.

Query Format:

:TRIGger:ALTernation:COUPling?□[<source>]

Return Format:

The query returns DC, AC, HF or LF.

Example:

:TRIGger:ALTernation:COUPling□DC, SOURceB

Set the couping mode as DC.

:TRIGger:ALTernation:COUPling?□SOURceB Return DC.

(16) :TRIGger:ALTernation:SENSitivity

Command Format:

:TRIGger:ALTernation:SENSitivity□<count>[,<source>]

Function:

The command sets the trigger sensitivity of alternation trigger, the count range is 0.1div~1div. The <source> may be SOURceA or SOURceB, and the source A and B are varying with to the current alternation channel.

Query Format:

:TRIGger:ALTernation:SENSitivity?□[<source>]

Return Format:

The query returns the value of trigger sensitivity, and the unit is div.

Example:

:TRIGger:ALTernation:SENSitivity□0.1, SOURceB

Set the trigger sensitivity as 01.

:TRIGger:ALTernation:SENSitivity?□SOURceB Return 1.00e-01.

5. Pattern Trigger

Pattern trigger identifies the trigger conditions by checking the specified code. The code is the logical combination of all the channels, each channel has three logical values: high (H), low (L) and ignore(X).

(1) :TRIGger:PATTern:PATTern

Command Format:

:TRIGger:PATTern:PATTern□<value>,<mask>,<ext source>[,<edge source>,<edge>]

Function:

The command sets the code pattern of signals.

- <value>: Code pattern values of 5 channels. It is a 16 bit unsigned integer, and High is 1, Low is 0; for example:HHLLH is 11001, translate into decimal system is 25.
- <mask>: Indicates the mask code of channel which is denoted by the low five digits of hexadecimal number, and separately indicates whether the mask code is 1 or 0. The relationship between <mask> and <value> is " and" , if the mask of a channel is 0, which denotes this channel is ineffective and the corresponding setting of oscilloscope is "X"; if the mask is 1, <value> will decide whether the channel is H or L.
- <ext source>:It is external trigger signal, and EXT5 is 1, EXT is 0;
- <edge source>:It is the current channel, its range: 0~4;
- <edge>: It is the code pattern of current channel.The rising <edge> is 1and the falling<edge> is 0.

Query Format:

:TRIGger:PATTern:PATTern?

Return Format:

The query returns the value, the mask, the ext source, the edge source and the edge. The value and the mask are expressed in decimal.

Example:

:TRIGGER:PATTERN:PATTERN□1,1,1,2,1 Set the code pattern.

:TRIGger:PATTern:PATTern? Return 1, 5, EXT5, Channel3, Positive.

Math Commands

Math Commands are used to display the result of adding, subtracting, multiplying and FFT operation for the signals from CH1, CH2, CH3 and CH4. The results can be measured by the grid and the cursor.

Math Command includes:

- :MATH:DISPlay

1. :MATH:DISPlay

Command Format:

:MATH:DISPlay□{{1|ON}|{0|OFF}}

Function:

The command sets Math waveform on or off.

Query Format:

:MATH:DISPlay?

Return Format:

The query returns 1 or 0, respectively indicates ON or OFF.

Example:

:MATH:DISPlay□ON Set Math waveform on.

:MATH:DISPlay? Return 1.

Waveform Commands

Waveform Commands are used to read the datum and parameters of waveform on the screen.

Waveform Commands include:

- :WAVeform:DATA?
- :WAVeform:FORMat
- :WAVeform:SOURce
- :WAVeform:WINMemsize?
- :WAVeform:YINCrement?
- :WAVeform:YORigin?
- :WAVeform:XINCrement?
- :WAVeform:XORigin?
- :WAVeform:STARtofroll?
- :WAVeform:ERASeofroll?
- :WAVeform:WINDowzoom?
- :WAVeform:SYSMemsize?
- :WAVeform:SCREENDATA?
- :WAVeform:MEMorydata?
- :WAVeform:TPOSition?
- :WAVeform:WPOSition?
- :WAVeform:PARAmeter?
- :WAVeform:XREFerence?
- :WAVeform:YREFerence?
- :WAVeform:DATA? XY <data>

1. :WAVeform:DATA?

Command Format:

:WAVeform:DATA?□[<source>]

Function:

The command is used to read the data of waveform on EMS memory. The <source> may be CHANnel1, CHANnel2, CHANnel3, CHANnel4 or MATH.

Return Format:

The query returns 1024 waveform datum.

Example:

:WAVeform:DATA?□CHANnel1

Read the data from CH1, return: 5.240000e+02, 4.760000e+02,…….

2. :WAVeform:FORMat

Command Format:

:WAVeform:FORMat□<value>

Function:

The command sets the format of waveform data. The <value> may be WORD, BYTE or ASCII.

Difference of WORD, BYTE and ASCII:

ASCII: Datum translate into character, returns ASCII values.

Eg: Waveform data is 1000, returns '1','0','0','0', a point correspond many bytes.

BYTE and ASCII: They will returns 8 bit and 16 bit values to use datum.

Eg:Waveform data is 1000, returns 1000 in decimal system, a point correspond one byte.

Query Format:

:WAVeform:FORMat?

Return Format:

The query returns WORD or BYTE, ASCII.

Example:

:WAVeform:FORMat□ASCII Set the data format as ASCII.

:WAVeform:FORMat? Return ASCII.

3. :WAVeform:SOURce

Command Format:

:WAVeform:SOURce□<source>

Function:

The command sets the source of waveform data which is going to check. The <source> may be CHANnel1, CHANnel2, CHANnel3, CHANnel4 or MATH.

Query Format:

:WAVeform:SOURce?

Return Format:

The query returns Channel1 or Channel2, Channel3, Channel4, MATH.

Example:

:WAVeform:SOURce□CHANnel2 Set the data source as channel 2.
:WAVeform:SOURce? Return Channel2.

4. :WAVeform:WINMemsize?

Command Format:

:WAVeform:WINMemsize?□[<source>]

Function:

The command queries the data length of specified soucce. The <source> may be CHANnel1, CHANnel2, CHANnel3, CHANnel4 or MATH.

Return Format:

The query returns the value of data length.

Example:

:WAVeform:WINMemsize?□CHANnel2 Return 1024.

5. :WAVeform:YINCrement?

Command Format:

:WAVeform:YINCrement?□[<source>]

Function:

The command queries the vertical scale of specified soucce. The <source> may be CHANnel1, CHANnel2, CHANnel3, CHANnel4 or MATH.
NOTE: returned value=scale value/25.

Return Format:
The query returns the value of vertical scale, and the unit is V.

Example:
:WAVeform:YINCrement?□CHANnel2 Return 4.00e+00.

6. :WAVeform:YORigin?

Command Format:
:WAVeform:YORigin?□[<source>]

Function:
The command queries the vertical offset of specified soucce. The <source> may be CHANnel1, CHANnel2, CHANnel3, CHANnel4 or MATH.

Return Format:
The query returns the value of vertical offset, and the unit is V.

Example:
:WAVeform:YORigin?□CHANnel2 Return -1.60e+01.

7. :WAVeform:XINCrement?

Command Format:
:WAVeform:XINCrement?□[<source>]

Function:
The command queries the interval of specified soucce. The <source> may be CHANnel1, CHANnel2, CHANnel3, CHANnel4 or MATH.

Return Format:
The query returns the value of interval, and the unit is s.

Example:
:WAVeform:XINCrement?□CHANnel2 Return 1.00e-03.

8. :WAVeform:XORigin?

Command Format:

:WAVeform:XORigin?□[<source>]

Function:

The command queries the time from origin to start of EMS memory of specified soucce. The <source> may be CHANnel1, CHANnel2, CHANnel3, CHANnel4 or MATH.

Return Format:

The query returns the value of time and the unit is s.

Example:

:WAVeform:XORigin?□CHANnel2 Return 2.00e-02.

9. :WAVeform:STARtofroll?

Command Format:

:WAVeform:STARtofroll?

Function:

The command queries the start point of roll display.

Return Format:

The query returns the coordinate of start point.

Example:

:WAVeform:STARtofroll? Return 0.

10. :WAVeform:ERASeofroll?

Command Format:

:WAVeform:ERASeofroll?

Function:

The command queries the start point of roll erase.

Return Format:

The query returns the coordinate of start point.

Example:
:WAVeform:ERASeofroll? Return 3.

11. :WAVeform:WINDowzoom?

Command Format:
:WAVeform:WINDowzoom?

Function:
The command queries the window width.

Return Format:
The query returns the value of window width, and the unit is pels.

Example:
:WAVeform:WINDowzoom? Return 300.

12. :WAVeform:SYSMemsize?

Command Format:
:WAVeform:SYSMemsize?

Function:
The command queries the system memory depth.

Return Format:
The query returns the value of memory depth.

Example:
:WAVeform:SYSMemsize? Return 8192.

13. :WAVeform:SCREENDATA?

Command Format:
:WAVeform:SCREENDATA?□[<source>]

Function:
The command queries the waveform data of screen. The <source> may be CHANnel1, CHANnel2, CHANnel3, CHANnel4 or MATH.

Return Format:

The query returns the value of screen, and the unit is V.

Example:

:WAVeform:SCREENDATA?□CHANNEL1

Query the screen data of channel1, return: 4.000000e+02, …….

14. :WAVeform:MEMorydata?

Command Format:

:WAVeform:MEMorydata?□[<source>]

Function:

The command queries the memory data of specified source. The <source> may be CHANnel1, CHANnel2, CHANnel3, CHANnel4 or MATH.

Return Format:

The query returns the value of memory data, and the unit is V.

Example:

:WAVeform:MEMorydata?□CHANNEL1

Query the memory data of channel 1, return 4.000000e+00, …….

15. :WAVeform:TPOSition?

Command Format:

:WAVeform:TPOSition?

Function:

The command queries the trigger position.

Return Format:

The query returns the axis of trigger position.

Example:

:WAVeform:TPOSition? Return 2048.

16. :WAVeform:WPOSition?

Command Format:

:WAVeform:WPOSition?

Function:

The command queries the position that the left of screen reaches the start point of memory.

Return Format:

The query returns the coordinate of position.

Example:

:WAVeform:WPOSition? Return 1696.

17. :WAVeform:PARAmeter?

Command Format:

:WAVeform:PARAmeter?<para>

Function:

The command queries the system draw parameters.

Return Format:

The query returns the system draw parameters, which is composed of the header of TMC (#800000xxx) and DSO_WAVE_PARAMETER.

18. :WAVeform:XREFerence?

Command Format:

:WAVeform:XREFerence?

Function:

The command queries the horizontal reference axis.

Return Format:

The query returns the value of reference axis.

Example:

:WAVeform:XREFerence? Return 0.

19. :WAVeform:YREFerence?

Command Format:

:WAVeform:YREFerence?

Function:

The command queries the vertical reference axis.

Return Format:

The query returns the value of reference axis.

Example:

:WAVeform:YREFerence? Return 99.

20. :WAVeform:DATA? XY <data>

Command Format;

:WAVeform:DATA? XY <data>

Function:

Request the waveform in XY mode. 2K, 1K in front is channel1, 1K in back is channel2.

Utility Commands

Utility Commands are used to set other system function, such as counter, system language, system beep and real-time clock.

Utility Commands include:

- :COUNter:ENABle
- :BEEP:ENABle
- :BEEP:ACTion
- :RTC
- :INFO:LANGuage

1. :COUNter:ENABle

Command Format:

:COUNter:ENABle□{{1|ON}|{0|OFF}}

Function:

The command sets the counter ON or OFF.

Query Format:

:COUNter:ENABle?

Return Format:

The query returns 1 or 0, respectively indicates ON or OFF.

Example:

:COUNter:ENABle□ON Set the counter on.

:COUNter:ENABle? Return 1.

2. :BEEP:ENABle

Command Format:

:BEEP:ENABle□{{1|ON}|{0|OFF}}

Function:

The command sets the system beeper ON or OFF.

Query Format:

:BEEP:ENABle?

Return Format:

The query returns 1 or 0, respectively indicates ON or OFF.

Example:

:BEEP:ENABle□ON Set the system beeper on.

:BEEP:ENABle? Return 1.

3. :BEEP:ACTion

Command Format:

:BEEP:ACTion

Function:
The command tests the system beeper.

4. :RTC

Command Format:
:RTC□<year>,<month>,<day>,<hour>,<minute>,<second>

Function:
The command sets the system time.

Query Format:
:RTC?

Return Format:
The query returns the Year, the Month, the day, the hour, the minite, and the second.

Example:
:RTC□2008,8,8,20,08,08 Set the system time as 08, 08, 08, 08, 08, 08pm.
:RTC? Return 2008, 8, 8, 20, 8, 8.

5. :INFO:LANGuage

Command Format:
:INFO:LANGuage□<cmd_lang>

Function:
The command sets the system language. The <cmd_lang> may be SIMPlifiedchinese, TRADitionalchinese, KOREan, JAPanese, ENGLish, FRENch, GERMan, ITALian, RUSSian, PORTuguese or SPANish.

Query Format:
:INFO:LANGuage?

Return Format:
The query returns Simplified Chinese or Traditional Chinese, Korean, Japanese, English, German, French, Italian, Russian, Portuguese, Spanish.

Example:
:INFO:LANGuage□SIMP Set the system language as simplified chinese.
:INFO:LANGuage? Return Simplified Chinese.

Storage Commands

Storage Commands are used to save and recall the waveform data and screen image.

Storage Commands include:

- :SAVERECALL:TYPE
- :SAVERECALL:LOCation
- :SAVERECALL:LOAD
- :SAVERECALL:SAVE
- :SAVE:IMAGe:STARt
- :SAVE:IMAGe:FACTors
- :SAVE:IMAGe:FORMat
- :SAVE:WAVeform:STARt
- :SAVE:SETup:STARt
- :SAVE:CSV:STARt
- :RECall:WAVeform:STARt
- :RECall:SETup:STARt

1. :SAVERECALL:TYPE

Command Format:

:SAVERECALL:TYPE□<type>

Function:

The command sets the data type for storage. The <type> may be WAVeform (waveform data) or SETups (data settings).

Query Format:

:SAVERECALL:TYPE?

Return Format:

The query returns WAVEFORMS or SETUPS.

Example:

:SAVERECALL:TYPE□WAVeforms

Set the storage type as waveform data.

:SAVERECALL:TYPE? Return WAVEFORM.

2. :SAVERECALL:LOCation

Command Format:

:SAVERECALL:LOCation□<location>

Function:

The command sets the storage location. The <location> may be 0~9.

Query Format:

:SAVERECALL:LOCation?

Return Format:

The query returns 0 or 1 ……9.

Example:

:SAVERECALL:LOCation□1 Set the storage location as the second.

:SAVERECALL:LOCation? Return 1.

3. :SAVERECALL:LOAD

Command Format:
:SAVERECALL:LOAD

Function:
The command recalls the waveform or setup data from internal flash according to storage type.

4. :SAVERECALL:SAVE

Command Format:
:SAVERECALL:SAVE

Function:
The command saves the waveform or setup to internal flash according to storage type.

5. :SAVE:IMAGe:STARt

Command Format:
:SAVE:IMAGe:STARt <file_spec>
NOTE: In commands, the file name must add double quotation marks.

Function:
The command saves the image. The <file_spec> is the file name and must be ASCII characters, and the length of file name must be less than 256 characters.

6. :SAVE:IMAGe:FACTors

Command Format:
:SAVE:IMAGe:FACTors□{{1|ON}|{0|OFF}}

Function:
The command sets the saving function of system parameters on or off. The function indicates to save a file which records all system parameters while saving image.

Query Format:
:SAVE:IMAGe:FACTors?
Return Format:
The query returns 1 or 0, respectively indicates ON or OFF.

Example:

:SAVE:IMAGe:FACTors□ON Save the system parameters.

:SAVE:IMAGe:FACTors? Return 1.

7. :SAVE:IMAGe:FORMat

Command Format:

:SAVE:IMAGe:FORMat□<format>

Function:

The command sets the format of saved image. The <format> may be 24bit real color (BMP|BMP24bit), 8bit bitmap (BMP8bit) or PNG (PNG).

Query Format:

:SAVE:IMAGe:FORMat?

Return Format:

The query returns BMP24bit, BMP8bit or PNG.

Example:

:SAVE:IMAGe:FORMat□BMP Set the format as 24 bit real color.

:SAVE:IMAGe:FORMat? Return BMP24bit.

8. :SAVE:WAVeform:STARt

Command Format:

:SAVE:WAVeform:STARt <file_spec>

NOTE: In external save, the file name must add double quotation marks.

Function:

The command starts the saving waveform function. If the waveforms are in internal flash, the <file_spec> is composed of integers among 0~9; if in external storage medium, the <file_spec> is composed of ASCII characters.

9. :SAVE:SETup:STARt

Command Format:

:SAVE:SETup:STARt <file_spec>

NOTE: In external save, the file name must add double quotation marks.

Function:

The command starts the saving setup function. If the waveforms are in internal flash, the <file_spec> is composed of integers among 0~9; if in external storage medium, the <file_spec> is composed of ASCII characters.

10. :SAVE:CSV:STARt

Command Format:

:SAVE:CSV:STARt <file_spec>]

NOTE: In external save, the file name must add double quotation marks.

Function:

The command sets the saving function of CSV file on. CSV file can be saved in external storage medium. The <file_spec> is the file name and must be ASCII characters, and the length of file name must be less than 256 characters.

11. :RECall:WAVeform:STARt

Command Format:

:RECAll:WAVeform:STARt <file_spec>

NOTE: In external save, the file name must add double quotation marks.

Function:

The command sets the recalling waveform function on. If the waveforms are in internal flash, the <file_spec> is composed of integers among 0~9; if in external storage medium, the <file_spec> is composed of ASCII characters.

12. :RECall:SETup:STARt

Command Format:

:RECAll:SETup:STARt <file_spec>

NOTE: In external save, the file name must add double quotation marks.

Function:

The command sets the recalling setup function on. If the waveforms are in internal flash, the <file_spec> is composed of integers among 0~9; if in external storage medium, the <file_spec> is composed of ASCII characters.

Mask Commands

Mask Commad are used to create and modify the rules for pass/fail test function.

Mask Commands include:

- :MASK:CREate
- :MASK:ENABle
- :MASK:X
- :MASK:Y
- :MASK:SOURce
- :MASK:OPERate
- :MASK:OUTPut
- :MASK:STOPonoutput
- :MASK:SAVE
- :MASK:LOAD
- :MASK:DOWNload
- :MASK:Upload
- :MASK:MSG

1. :MASK:CREate

Command Format:

:MASK:CREate

Function:

The command creates the rule of passing test.

2. :MASK:ENABle

Command Format:

:MASK:ENABle□{{1|ON}|{0|OFF}}

Function:

The command sets the state of passing test as ON or OFF.

Query Format:

:MASK:ENABle?

Return Format:

The query returns 1 or 0, respectively indicates ON or OFF.

Example:

:MASK:ENABle□ON Set passing test on.

:MASK:ENABle? Return 1.

3. :MASK:X

Command Format:

:MASK:X□<x>

Function:

The command sets the rule of testing X direction. The <x> is 0.04div~4div.

Query Format:

:MASK:X?

Return Format:

The query returns the x value, and the unit is div.

Example:

:MASK:X□1 Set the X direction rule as 1div.

:MASK:X? Return 1.000000.

4. :MASK:Y

Command Format:

:MASK:Y□<y>

Function:

The command sets the rule of testing Y direction. The <y> is 0.04div~4div.

Query Format:

:MASK:Y?

Return Format:

The query returns the y value, and the unit is div.

Example:

:MASK:Y□1 Set the Y direction rule as 1div.

:MASK:Y? Return 1.000000.

5. :MASK:SOURce

Command Format:

:MASK:SOURce□<source>

Function:

The command sets the passing test source. The <source> may be CHANnel1, CHANnel2, CHANnel3 or CHANnel4.

Query Format:

:MASK:SOURce?

Return Format:

The query returns CHAN1 or CHAN2, CHAN3, CHAN4.

Example:

:MASK:SOURce□CHANnel1 Set the passing test source as channel 1.

:MASK:SOURce? Return CHAN1.

6. :MASK:OPERate

Command Format:

:MASK:OPERate□<opt>

Function:

The command sets the function of passing test run or stop. The <opt> may be RUN or STOP.

Query Format:

:MASK:OPERate?

Return Format:

The query returns RUN or STOP.

Example:

:MASK:OPERate□RUN	Set the operation of passing test run.
:MASK:OPERate?	Return RUN.

7. :MASK:OUTPut

Command Format:

:MASK:OUTPut□<output>

Function:

The command sets the output mode of passing test. The <output> may be FAIL, PASS, FAIL_SOUND or PASS_SOUND.

NOTE: PASS SOUND is effective when sound setting is on.

Query Format:

:MASK:OUTPut?

Return Format:

The query returns FAIL or PASS, FAIL_SOUND, PASS_SOUND.

Example:

:MASK:OUTPut□PASS	Set the ouput mode of passing test as pass.
:MASK:OUTPut?	Return PASS.

8. :MASK:STOPonoutput

Command Format:

:MASK:STOPonoutput□{{1|ON}|{0|OFF}}

Function:

The command sets the output stop mode of passing test ON or OFF.

Query Format:

:MASK:STOPonoutput?

Return Format:

The query returns 1 or 0, respectively indicates ON or OFF.

Example:

:MASK:STOPonoutput□ON Set the output stop mode of passing test on.

:MASK:STOPonoutput? Return 1.

9. :MASK:SAVE

Command Format:

:MASK:SAVE

Function:

The command saves the rule of passing test.

10. :MASK:LOAD

Command Format:

:MASK:LOAD

Function:

The command loads the rule of passing test.

11. :MASK:DOWNload

Command Format:

:MASK:DOWNload□<filename>

Function:

The command download the test rule to the external storage equipment, and the <filename> must be ASCII character.

NOTE: In commands, the file name must add double quotation marks.

12. :MASK:Upload

Command Format:

:MASK:Upload□<filename>

Function:

The command uploads the test rule from the external storage equipment, and the <filename> must be ASCII character.

NOTE: In commands, the file name must add double quotation marks.

13. :MASK:MSG

Command Format:

:MASK:MSG□{{1|ON}|{0|OFF}}

Function:

The command sets the prompt information function of passing test ON or OFF.

Query Format:

:MASK:MSG?

Return Format:

The query returns 1 or 0, respectively indicates ON or OFF.

Example:

:MASK:MSG□ON Set the prompt information function on.

:MASK:MSG? Return 1.

Cursor Commands

Cursor Commands are used to set cursor parameters to measure manually and automatically and track the waveform data.

Cursor Commands include:

- :CURSor:MODE
- :CURSor:MANUal:TYPE
- :CURSor:MANUal:SOURce
- :CURSor:MANUal:CURAX
- :CURSor:MANUal:CURAY
- :CURSor:MANUal:CURBX
- :CURSor:MANUal:CURBY
- :CURSor:TRACk:SOURceA
- :CURSor:TRACk:SOURceB
- :CURSor:TRACk:CURA
- :CURSor:TRACk:CURB

1. :CURSor:MODE

Command Format:

:CURSor:MODE□<mode>

Function:

The command sets the cursor mode. The <mode> may be CLOSe, MANUal, TRACk or MEASure (measure automatically).

Query Format:

:CURSor:MODE?

Return Format:

The query returns CLOSE or MANUAL, TRACK, MEASURE.

Example:

:CURSor:MODE□TRACK Set the cursor mode as track.
:CURSor:MODE? Return TRACK.

2. :CURSor:MANUal:TYPE

Command Format

:CURSor:MANUal:TYPE□<type>

Function:

The commad sets the cursor type of manual cursor. The <type> may be TIME or AMPlitude.

Query Format:

:CURSor:MANUal:TYPE?

Return Format:

The query returns Time or Amplitude.

Example:

:CURSor:MANUal:TYPE□TIMe Set the cursor type of manual cursor as time.
:CURSor:MANUal:TYPE? Return Time.

3. :CURSor:MANUal:SOURce

Command Format:
:CURSor:MANUal:SOURce□<source>

Function:
The command sets the cursor source of manual cursor. The <source> may be CHANnel1, CHANnel2, CHANnel3, CHANnel4 or MATH.

Query Format:
:CURSor:MANUal:SOURce?

Return Format:
The query returns Channel1 or Channel2, Channel3, Channel4, Math.

Example:
:CURSor:MANUal:SOURce□CHANnel1
Set the curosor source of manual cursor as channel 1.
:CURSor:MANUal:SOURce? Return Channel1.

4. :CURSor:MANUal:CURAX

Command Format:
:CURSor:MANUal:CURAX□<value>

Function:
The command sets the AX position of manual cursor. The <value> range is 4~297.

Query Format:
:CURSor:MANUal:CURAX?

Return Format:
The query returns the value of AX position.

Example:
:CURSor:MANUal:CURAX□100 Set the AX position of manual cursor as 100.
:CURSor:MANUal:CURAX? Return 100.

5. :CURSor:MANUal:CURAY

Command Format:
:CURSor:MANUal:CURAY□<value>

Function:
The command sets the AY position of manual cursor. The <value> range is 4~194.

Query Format:
:CURSor:MANUal:CURAY?

Return Format:
The query returns the value of AY position.

Example:
:CURSor:MANUal:CURAY□100 Set the AY position of manual cursor as 100.
:CURSor:MANUal:CURAY? Return 100.

6. :CURSor:MANUal:CURBX

Command Format:
:CURSor:MANUal:CURBX□<value>

Function:
The command sets the BX position of manual cursor. The <value> range is 4~297.

Query Format:
:CURSor:MANUal:CURBX?

Return Format:
The query returns the value of BX position.

Example:
:CURSor:MANUal:CURBX□100 Set the BX position of manual cursor as 100.
:CURSor:MANUal:CURBX? Return 100.

7. :CURSor:MANUal:CURBY

Command Format:
:CURSor:MANUal:CURBY□<value>

Function:

The command sets the BY position of manual cursor. The <value> range is 4~194.

Query Format:

:CURSor:MANUal:CURBY?

Return Format:

The query returns the value of BY position.

Example:

:CURSor:MANUal:CURBY□100 Set the BY position of manual cursor as 100.

:CURSor:MANUal:CURBY? Return 100.

8. :CURSor:TRACk:SOURceA

Command Format:

:CURSor:TRACk:SOURceA□<source>

Function:

The command sets the signal source A of track cursor. The <source> may be CHANnel1, CHANnel2, CHANnel3, CHANnel4, MATH or NONE.

Query Format:

:CURSor:TRACk:SOURceA?

Return Format:

The query returns Channel1 or Channel2, Channel3, Channel4, Math, None.

Example:

:CURSor:TRACk:SOURceA□CHANnel1

Set the signal source A of track cursor as channel 1.

:CURSor:TRACk:SOURceA? Return Channel1.

9. :CURSor:TRACk:SOURceB

Command Format:

:CURSor:TRACk:SOURceB□<source>

Function:

The command sets the signal source B of track cursor. The <source> may be CHANnel1, CHANnel2, CHANnel3, CHANnel4, MATH or NONE.

Query Format:

:CURSor:TRACk:SOURceB?

Return Format:

The query returns Channel1 or Channel2, Channel3, Channel4, Math, None.

Example:

:CURSor:TRACk:SOURceB□CHANnel1

Set the signal source B of track cursor as channel 1.

:CURSor:TRACk:SOURceB? Return Channel1.

10. :CURSor:TRACk:CURA

Command Format:

:CURSor:TRACk:CURA□<value>

Function:

The command sets the position of track cursor A. The <value> range is 4~297.

Query Format:

:CURSor:TRACk:CURA?

Return Format:

The query returns the position of cursor A.

Example:

:CURSor:TRACk:CURA□100 Set the position of track cursor A as 100.

:CURSor:TRACk:CURA? Return 100.

11. :CURSor:TRACk:CURB

Command Format:

:CURSor:TRACk:CURB□<value>

Function:

The command sets the position of track cursor B. The <value> range is 4~297.

Query Format:

:CURSor:TRACk:CURB?

Return Format:

The query returns the position of cursor B.

Example:

:CURSor:TRACk:CURB□100 Set the position of track cursor B as 100.

:CURSor:TRACk:CURB? Return 100.

Other Commands

There are some other commands except for above commands: clear the system error information, download the system setup data, enable and disable the AUTO key and enable and disable the Measure key.

The commands include:

- :SYSTem:ERRor
- :SYSTem:ERRor?
- :SYSTem:SETup
- :SYSTem:SETup?
- :AUToscale:DISable
- :AUToscale:ENable
- :AUToscale?

1. :SYSTem:ERRor

Command Format:

:SYSTem:ERRor

Function:

The command clears the queue of error information.

Query Format:

:SYSTem:ERRor?

Return Format:

If there are errors, return 0, No error; or else return Undefine header.

Example:

:SYSTem:ERRor? Return 0, No error.

2. :SYSTem:ERRor?

Command Format:

:SYSTem:ERRor?

Function:

Return the last error message.

Return Format:

0，No error

3. :SYSTem:SETup

Command Format:

:SYSTem:SETup□<value>

Function:

The command downloads the system setup data. The <value> is a binary data that meets IEEE 488.2 # format.

Query Format:

:SYSTem:SETup?

Return Format:

The query returns the value of system setup data.

4. :SYSTem:SETup?

Command Format:

:SYSTem:SETup?

Function:

Return the data of system SETUP.

5. :AUToscale:DISable

Command Format:

:AUTOscale:DISable

Function:

The command disables the AUTO key, and forbids users setting automatically.

6. :AUToscale:ENable

Command Format:

:AUToscale:ENable

Function:

The command enables the AUTO key, and allows users to set automatically.

7. :AUToscale?

Command Format:

:AUToscale?

Function:

Return the AUTOSCALE state.

Chapter 3 Programming Examples

This chapter lists some application examples of **RIGOL** DS1000B commands (C language). Users can compare or refer to them to get further understanding for the usage of commands.

The contents between "/*"and "*/" before every command are explanations which intends to help users understand instead of the contents of command.

Before operating, please make sure that the related test equipment and the device under test have been already connected correctly.

Example 1 The Preparation for Programming

```
#include <stdlib.h>
#include <visa.h>

/*Define the equipmant address*/
/*LAN*/
#define RESOURCE "TCPIP0::172.16.3.73::inst0::INSTR"

/*USB*/
#define RESOURCE "USB0::6833::1160::DS1BA103700043:0:INSTR"

/*Define the equipment ID*/
ViSession g_viDefault;
ViSession g_vi;

void main
{
    /*Open the equipment ID*/
    viOpenDefaultRM(&g_viDefault);
    viOpen(g_viDefault,RESOURCE,VI_NULL,VI_NULL,&vi);

    /*Clear the interface*/
    viClear(g_vi);

    /*Implement functions*/
    .......

    /*Close the equipmant ID*/
    viClose(g_vi);
    viClose(g_viDefault);
}
```

Example 2 Setting the basic measurement

```
/*Save the character string of equipment identifier the *IDN command returned*/
char s8IDNString[256];

void BasicSetup(void)
{
    /*Firstly acquire the equipment identifier*/
    viQueryf(g_vi,"*IDN?\n","%t",s8IDNString);

    /*Send the command :AUTO, and the equipment sets automatically according to the current signal to display best*/
    viPrintf(g_vi,":AUTO\n");

    /*Send the command :CHANnel<n>:PROBe to set the probe attenuation factor*/
    viPrintf(g_vi,":CHAN1:PROB 10X\n");

    /*Send the command :CHANnel<n>:OFFSet to set the vertical offset*/
    viPrintf(g_vi,":CHAN2:OFFS 20\n");

    /*Send the command :TIMebase[:MAIN]:SCALe to set the horizontal scale*/
    viPrintf(g_vi,":TIM:SCAL 1e-8\n");

    /*Send the command :TIMebase[:MAIN]:OFFSet to set the horizontal timebase*/
    viPrintf(g_vi,":TIM:OFFS 2e-6\n");

    /*Send the command :TRIGger:MODE to set the trigger mode*/
    viPrintf(g_vi,"TRIG:MODE EDGE\n");

    /*Send the command :TRIGger:EDGE:SLOPe to set the edge type of edge trigger*/
    viPrintf(g_vi,"TRIG:EDGE:SLOP POS\n");
}
```

Example 3 Storaging and Setting the File

```
#define BUFFER_SETUP_SIZE     1636

void SetupUpload()
{
    BYTE *bufSetup = new BYTE[BUFFER_SETUP_SIZE];
    FILE *fp;

    /*Judge the buffer validity */
    if (bufSetup == NULL)
    {
        return;
    }

    /*Make a statement for the VISA status*/
    ViStatus sta;

    /*Send the command :SYSTEM:SETUP? to save the file to buffer*/
    sta=viQueryf(g_vi,":SYSTEM:SETUP?\n","%#b\n",BUFFER_SETUP_SIZE,
bufSetup);

    /*Judge whether the command is sent successfully*/
    if(sta != VI_SUCCESS)
    {
        return;
    }

    /*Save the setup file to PC*/
    fp = fopen("C:\\scope\\setup.dat","wb");
    if(fwrite(bufSetup,sizeof(BYTE),BUFFER_SETUP_SIZE,fp)!=BUFFER_SETUP
_SIZE)
    {
        fclose(fp);
        return ;
    }

    fclose(fp);
}
```

```
void SetupDownLoad()
{
    BYTE *bufSetup = new BYTE[BUFFER_SETUP_SIZE];
    FILE *fp;

    /*Judge the buffer validity */
    if (bufSetup == NULL)
    {
        return;
    }

    /*Open the setup file*/
    fp = fopen("C:\\scope\\setup.dat","rb");
    if(fread(bufSetup,sizeof(BYTE),BUFFER_SETUP_SIZE,fp)!=BUFFER_SETUP
_SIZE)
    {
        fclose(fp);
        return ;
    }

    fclose(fp);

    /*Make a statement for VISA 32bit data type*/
    ViUInt32 ret

    /*Send the :SYSTEM:SETUP command to set the system parameters
    by the setup file*/
    viPrintf(g_vi,":SYSTEM:SETUP #8%08d",BUFFER_SETUP_SIZE);

    if (viBufWrite(g_vi,bufSetup,BUFFER_SETUP_SIZE,&ret) == VI_SUCCESS)
    {
        viPrintf(g_vi,"\n");
    }

}
```

Example 4 Reading the Data of Screen Image

```
#define BUFFER_IMAGE_SIZE     320*240+256*4+54+1;

void ImageUpload()
{
    BYTE *bufImage = new BYTE[BUFFER_IMAGE_SIZE];
    FILE *fp;

    long headLen    = 10;
    long sizeImage = BUFFER_IMAGE_SIZE;

    /*Judge the buffer validity*/
    if (bufImage == NULL)
    {
        return;
    }

    /*Make a statement for the VISA status*/
    ViStatus staDISP;
    ViStatus sta488;
    ViStatus staImage;

    /*Make a statement for IEEE488.2 standard data header*/
    BYTE head488[11] = {0};

    /*Send the :DISP:DATA?\n command to read the data of screen
    image*/
    sta         = viPrintf(g_vi,":DISP:DATA?\n");
    sta488     = viRead(g_vi,head488,headLen,&headLen);
    staImage = viRead(g_vi,bufImage,sizeImage,&sizeImage);

    if (sta == VI_SUCCESS && sta488 == VI_SUCCESS && staImage ==
VI_SUCCESS)
    {
        fp = fopen("C:\\scope\\screen.bmp","wb");
        if          (fread(bufImage,sizeof(BYTE),BUFFER_IMAGE_SIZE,fp)          !=
BUFFER_IMAGE_SIZE)
        {
```

```
                fclose(fp);
                return ;
            }

            fclose(fp);
        }
}
```

Command Quick Reference

*IDN? 2-3
*RST 2-4
*LRN? 2-4
*OPC? 2-5

A

:ACQuire:TYPE 2-34
:ACQuire:MODE 2-34
:ACQuire:AVERages2-35
:ACQuire:SRATe? 2-35
:AUTO 2-3
:AUToscale:DISable2-111
:AUToscale:ENable 2-111
:AUToscale? 2-111

B

:BEEP:ENABle 2-89
:BEEP:ACTion 2-89

C

:CHANnel<n>:BWLimit 2-43
:CHANnel<n>:COUPling 2-43
:CHANnel<n>:DISPlay 2-44
:CHANnel<n>:INVert 2-44
:CHANnel<n>:OFFSet 2-44
:CHANnel<n>:PROBe 2-45
:CHANnel<n>:SCALe 2-45
:CHANnel<n>:FILTer 2-46
:CHANnel<n>:MEMoryDepth?2-47
:CHANnel<n>:UNITs2-47
:CHANnel<n>:VERNier 2-48
:COUNter:ENABle 2-89
:CURSor:MODE 2-103
:CURSor:MANUal:TYPE 2-103
:CURSor:MANUal:SOURce 2-103
:CURSor:MANUal:CURAX 2-104
:CURSor:MANUal:CURAY 2-104
:CURSor:MANUal:CURBX 2-105
:CURSor:MANUal:CURBY 2-105
:CURSor:TRACk:SOURceA 2-106
:CURSor:TRACk:SOURceB 2-106
:CURSor:TRACk:CURA 2-107
:CURSor:TRACk:CURB 2-107

D

:DISPlay:DATA? 2-41
:DISPlay:TYPE 2-37
:DISPlay:GRID 2-37
:DISPlay:PERSist 2-38
:DISPlay:MNUDisplay 2-38
:DISPlay:MNUStatus 2-39
:DISPlay:SCReen 2-39
:DISPlay:CLEar 2-39
:DISPlay:BRIGhtness 2-40
:DISPlay:INTensity 2-40

F

:FORCetrig 2-4

I

:INFO:LANGuage 2-90

K

:KEY:LOCK 2-8
:KEY:RUN 2-8
:KEY:AUTO 2-8
:KEY:SINGLe 2-8
:KEY:CH1 2-9

:KEY:CH2 2-9
:KEY:CH3 2-9
:KEY:CH4 2-9
:KEY:MATH 2-9
:KEY:REF 2-10
:KEY:F1 2-10
:KEY:F2 2-10
:KEY:F3 2-10
:KEY:F4 2-11
:KEY:F5 2-11
:KEY:MNU 2-11
:KEY:MEASure 2-11
:KEY:CURSor 2-11
:KEY:ACQuire 2-12
:KEY:DISPlay 2-12
:KEY:STORage 2-12
:KEY:UTILity 2-12
:KEY:MNUTIME 2-12
:KEY:TrigMENU 2-13
:KEY:Trig%50 2-13
:KEY:TrigMODE 2-13
:KEY:TrigFORC 2-13
:KEY:HELP 2-13
:KEY:QUICKMEASure 2-14
:KEY:QUICKPRINt 2-14
:KEY:FUNC_Z 2-14
:KEY:FUNC_INC 2-14
:KEY:FUNC_DEC 2-14
:KEY:CH1_VOLT_INC 2-15
:KEY:CH1_VOLT_DEC 2-15
:KEY:CH1_VOLT_Z 2-15
:KEY:CH1_POS_INC 2-15
:KEY:CH1_POS_DEC 2-15
:KEY:CH1_POS_Z 2-16
:KEY:CH2_VOLT_INC 2-16
:KEY:CH2_VOLT_DEC 2-16
:KEY:CH2_VOLT_Z 2-16
:KEY:CH2_POS_INC 2-16
:KEY:CH2_POS_DEC 2-17
:KEY:CH2_POS_Z 2-17
:KEY:CH3_VOLT_INC 2-17
:KEY:CH3_VOLT_DEC 2-17
:KEY:CH3_VOLT_Z 2-17
:KEY:CH3_POS_INC 2-18
:KEY:CH3_POS_DEC 2-18
:KEY:CH3_POS_Z 2-18
:KEY:CH4_VOLT_INC 2-18
:KEY:CH4_VOLT_DEC 2-18
:KEY:CH4_VLOT_Z 2-19
:KEY:CH4_POS_INC 2-19
:KEY:CH4_POS_DEC 2-19
:KEY:CH4_POS_Z 2-19
:KEY:TIME_INC 2-19
:KEY:TIME_DEC 2-20
:KEY:TIME_Z 2-20
:KEY:TIME_POS_INC 2-20
:KEY:TIME_POS_DEC 2-20
:KEY:TIME_POS_Z 2-20
:KEY:TRIG_LEVEL_INC 2-21
:KEY:TRIG_LEVEL_DEC 2-21
:KEY:TRIG_LEVEL_Z 2-21

M

:MASK:CREate 2-97
:MASK:ENABle 2-97
:MASK:X 2-97
:MASK:Y 2-98
:MASK:SOURce 2-98
:MASK:OPERate 2-99
:MASK:OUTPut 2-99
:MASK:STOPonoutput 2-100
:MASK:SAVE 2-100
:MASK:LOAD 2-100
:MASK:DOWNload 2-100
:MASK:Upload 2-101
:MASK:MSG 2-101
:MATH:DISPlay 2-78
:MEASure:CLEar 2-23
:MEASure:VPP? 2-23
:MEASure:VMAX? 2-23

:MEASure:VMIN? 2-23
:MEASure:VAMPlitude? 2-24
:MEASure:ENABle2-24
:MEASure:DISable2-24
:MEASure? 2-24
:MEASure:VTOP? 2-24
:MEASure:VBASe? 2-24
:MEASure:VAVerage? 2-25
:MEASure:VRMS? 2-25
:MEASure:OVERshoot? 2-25
:MEASure:PREShoot? 2-26
:MEASure:FREQuency? 2-26
:MEASure:RISetime? 2-26
:MEASure:FALLtime? 2-26
:MEASure:PERiod? 2-27
:MEASure:PWIDth? 2-27
:MEASure:NWIDth? 2-27
:MEASure:PDUTycycle? 2-28
:MEASure:NDUTycycle? 2-28
:MEASure:PDELay? 2-28
:MEASure:NDELay? 2-29
:MEASure:PPHAse? 2-29
:MEASure:NPHAse? 2-29
:MEASure:TOTal 2-30
:MEASure:SOURce 2-30
:MEASure:DELAYSOUR 2-30
:MEASure:PHAseSOUR 2-30
:MEASure:ENABle2-32
:MEASure:DISable2-32

R

:RECall:WAVeform:STARt 2-95
:RECall:SETup:STARt 2-95
:RTC 2-90
:RUN 2-3

S

:SAVERECALL:TYPE 2-92
:SAVERECALL:LOCation 2-92
:SAVERECALL:LOAD 2-92
:SAVERECALL:SAVE 2-93
:SAVE:IMAGe:STARt 2-93
:SAVE:IMAGe:FACTors 2-93
:SAVE:IMAGe:FORMat 2-94
:SAVE:WAVeform:STARt 2-94
:SAVE:SETup:STARt 2-94
:SAVE:CSV:STARt 2-95
:STOP 2-3
:SYSTem:ERRor2-110
:SYSTem:ERRor?2-110
:SYSTem:SETup2-110
:SYSTem:SETup?2-111

T

:TIMebase:MODE 2-50
:TIMebase:DELayed:OFFSet 2-50
:TIMebase:DELayed:SCALe 2-51
:TIMebase[:MAIN]:OFFSet 2-51
:TIMebase[:MAIN]:SCALe 2-52
:TIMebase:FORMat 2-53
:Trig%50 2-4
:TRIGger:MODE 2-55
:TRIGger<mode>:SOURce 2-55
:TRIGger<mode>:LEVel 2-56
:TRIGger<mode>:SWEep 2-56
:TRIGger:COUPling 2-57
:TRIGger:HOLDoff 2-57
:TRIGger:STATus 2-58
:TRIGger:SENSitivity 2-58
:TRIGger:HFREject 2-59
:TRIGger:EDGE:SLOPe 2-61
:TRIGger:PULSe:MODE 2-62
:TRIGger:PULSe:WIDTh 2-62
:TRIGger:VIDEO:MODE 2-64
:TRIGger:VIDEO:POLarity 2-64
:TRIGger:VIDEO:STANdard 2-65
:TRIGger:VIDEO:LINE 2-65
:TRIGger:ALTernation:SOURce 2-66
:TRIGger:ALTernation:CURRentSOURce 2-66
:TRIGger:ALTernation:TYPE 2-67

:TRIGger:ALTernation:TimeSCALe 2-67
:TRIGger:ALTernation:TimeOFFSet 2-68
:TRIGger:ALTernation:LEVel 2-69
:TRIGger:ALTernation:EDGE:SLOPe 2-69
:TRIGger:ALTernation:PULSe:MODE 2-70
:TRIGger:ALTernation:PULSe:TIME 2-70
:TRIGger:ALTernation:VIDEO:POLarity 2-71
:TRIGger:ALTernation:VIDEO:STANdard 2-71
:TRIGger:ALTernation:VIDEO:MODE 2-72
:TRIGger:ALTernation:HOLDoff 2-73
:TRIGger:ALTernation:VIDEO:LINE 2-73
:TRIGger:ALTernation:COUPling 2-74
:TRIGger:ALTernation:SENSitivity 2-74
:TRIGger:PATTern:PATTern 2-76

W

:WAVeform:DATA? 2-80
:WAVeform:FORMat 2-80
:WAVeform:SOURce 2-81
:WAVeform:WINMemsize? 2-81
:WAVeform:YINCrement? 2-81
:WAVeform:YORigin? 2-82
:WAVeform:XINCrement? 2-82
:WAVeform:YINCrement? 2-81
:WAVeform:YORigin? 2-82
:WAVeform:XINCrement? 2-82
:WAVeform:XORigin? 2-83
:WAVeform:STARtofroll? 2-83
:WAVeform:ERASeofroll? 2-83
:WAVeform:WINDowzoom? 2-84
:WAVeform:SYSMemsize? 2-84
:WAVeform:SCREENDATA? 2-84
:WAVeform:MEMorydata? 2-85
:WAVeform:TPOSition? 2-85
:WAVeform:WPOSition? 2-85
:WAVeform:PARAmeter? 2-86
:WAVeform:XREFerence? 2-86
:WAVeform:YREFerence? 2-86